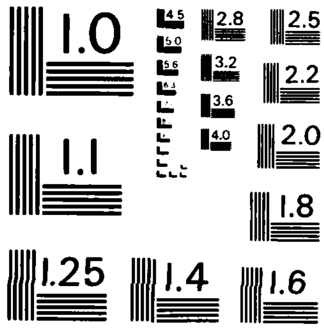
AD-A161 349 ROBUST ADAPTIVE CONTROL(U) INTEGRATED SYSTEMS INC PALO ALTO CA R L KOSUT 19 SEP 85 ISI-61 AFOSR-TR-85-0968 F49620-84-C-0085 1/1

UNCLASSIFIED F/G 9/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

integrated systems inc.

AFOSR-TR- 9 8

AD-A161 349

ROBUST ADAPTIVE CONTROL

FINAL REPORT

PREPARED BY:

ROBERT L. KOSUT

PREPARED FOR:

AFOSR
BUILDING 410
BOLLING AFB
WASHINGTON DC 20332

ATTN: DR. MARK JACOBS

PREPARED UNDER:

CONTRACT F49620-84-C-0085

ISI REPORT 61 SEPTEMBER 1985

415-853-8400 Telex 559631
101 University Avenue
Palo Alto, CA 94301-1695

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFOSR-TR- 3	2. GOVT ACCESSION NO. AD-A161	3. RECIPIENT'S CATALOG NUMBER 349
4. TITLE (and Subtitle) ROBUST ADAPTIVE CONTROL		5. TYPE OF REPORT & PERIOD COVERED FINAL 9/15/84 - 9/15/85
		6. PERFORMING ORG. REPORT NUMBER ISI-61
7. AUTHOR(s) Dr. Robert L. Kosut		8. CONTRACT OR GRANT NUMBER(s) F49620-84-C-0085
9. PERFORMING ORGANIZATION NAME AND ADDRESS INTEGRATED SYSTEMS, INC. 101 University Avenue Palo Alto, CA 94301		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304 A1
11. CONTROLLING OFFICE NAME AND ADDRESS AFOSR/NM Bldg 410 Bolling AFB DC 20332-6448		12. REPORT DATE 9-19-85
		13. NUMBER OF PAGES 75
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) SAME AS ABOVE		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

SAME AS ABOVE

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEE ATTACHED PAGE FOR BLOCK 20 - ABSTRACT

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

MATERIAL INSPECTION AND RECEIVING REPORT

1. PROC. INSTRUMENT IDEN (CONTRACT)	ORDER NO.	6. INVOICE NO. / DATE	7. PAGE	OF	8. ACCEPTANCE POINT
F49620-84-C-0085			1	1	D

2. SHIPMENT NO.	3. DATE SHIPPED	4. B/L TCN	5. DISCOUNT TERMS
ISI0001Z	85 SEPT 19		None

9. PRIME CONTRACTOR CODE FSC 4U288

INTEGRATED SYSTEMS, INC.
101 University Avenue, Suite 300
Palo Alto, CA 94301

10. ADMINISTERED BY CODE S0507A

DCASMA, San Francisco
1250 Bayhill Drive
San Bruno, CA 94066

11. SHIPPED FROM (If other than 9) CODE FOB

12. PAYMENT WILL BE MADE BY CODE S0506A

DCASR Los Angeles
P.O. Box 45011
Los Angeles, CA 90045

13. SHIPPED TO CODE

Dr. Mark Jacobs
Directorate of Mathematical & Information Sciences
AFOSR, Building 410
Bolling AFB, Washington DC 20332

14. MARKED FOR CODE

15. ITEM NO.	16. STOCK/PART NO. DESCRIPTION (Indicate number of shipping containers, type of container, container number)	17. QUANTITY SHIP'D/REC'D*	18. UNIT	19. UNIT PRICE	20. AMOUNT
001	Final Report: ROBUST ADAPTIVE CONTROL	6 Copies	Job		

21. PROCUREMENT QUALITY ASSURANCE

A. ORIGIN
PQA ACCEPTANCE of listed items has been made by me or under my supervision and they conform to contract, except as noted herein or on supporting documents.

DATE SIGNATURE OF AUTH GOVT REP
TYPED NAME AND OFFICE

B. DESTINATION
PQA ACCEPTANCE of listed items has been made by me or under my supervision and they conform to contract, except as noted herein or on supporting documents.

DATE SIGNATURE OF AUTH GOVT REP
TYPED NAME AND TITLE

22. RECEIVER'S USE
Quantities shown in column 17 were received in apparent good condition except as noted.

DATE RECEIVED SIGNATURE OF AUTH GOVT REP
TYPED NAME AND OFFICE

*If quantity received by the Government is the same as quantity shipped, indicate by (✓) mark, if different, enter actual quantity received below quantity shipped and encircle.

23. CONTRACTOR USE ONLY

The final report shows all results obtained during the course of the research activity.

Michael G. Lyons
Vice President

BLOCK 20 - ABSTRACT

ABSTRACT

A-1 23

This project addresses the problems involved in developing an interactive software system that integrates adaptive and nonlinear control design procedures with a real-time processor for implementation. The proposed system implements complex control laws in the laboratory for rapid testing evaluation and tuning. Recent hardware development and software methodologies form the basis of computer-aided control system design (CACSD) products. The two basic elements defined in the study are:

(1) A user friendly CACSD software package for product development and validation of control laws using simulation models

(2) A real-time hardware system which can automatically implement control laws designed by the CACSD software package, without real-time programming.

The products proposed under this project will significantly reduce engineering development time. Many new applications will occur as more and more improved CAE tools are available and the designed control laws can be rapidly verified in the laboratory. The hardware will be capable of implementing optimal trajectories allowing more economical operation of process plants and automated manufacturing lines. Advanced control, guidance, and estimation methods will be introduced into operational systems rapidly since development times and risks will be reduced. Easy to use laboratory test tools will eventually change policymaker's decisions on appropriate funding levels for new complex systems development.

Table of Contents

SECTION 1
PROJECT OVERVIEW

1.1 NEEDS IN COMPUTER-AIDED CONTROL SYSTEM DESIGN (CASD)

The high performance requirements and complexity of modern weapons, engines, and chemical processes have made it necessary to use sophisticated control logic. In addition, systems must operate efficiently over a wide range of conditions, providing the incentive to develop an "intelligent/adaptive" feedback controller. The controller must be able to implement in real-time and on-line most design functions now performed off-line by the control engineer. To realize this aim, both a theory of stability and performance of such inherently nonlinear controls is essential. It must also be possible to implement such complex control laws in the laboratory for rapid testing evaluation, and tuning. Recent advances in semiconductor technology and architectures have made it practical to rapidly achieve such implementation.

Two principal capabilities are needed for rapid development and testing of control laws:

(1) a user-friendly control design computer-aided-engineering (CAE) software package for development and validation of control laws using simulation models, and

(2) a real-time hardware system which can automatically implement control laws designed by the CAE software package. No real-time programming should be necessary.

Unfortunately, a user-friendly software system which can work in conjunction with a compatible real-time processor is currently not available. Typically, practical synthesis is limited to a trial and error time consuming procedure. With the help of an interactive software system that integrates adaptive/ nonlinear control design procedures and real-time processor implementation, the overall set-up time in the control design cycle can be reduced significantly.

The objective of this Phase I activity is to define a hardware and software real-time system with capability to demonstrate successful nonlinear and adaptive control for aerospace systems in laboratory hardware-in-the-loop testing. The Phase I study has established the usefulness and feasibility of developing an integrated CACSD system.

1.2 SUMMARY OF PHASE I RESEARCH - THEORY DEVELOPMENT AND FEASIBILITY STUDY

The Phase I research (period of performance is 15 September 1984 to 14 March 1985) obtained quantitative measures for the performance and stability of adaptive control systems. Results to date have established the following:

(1) Conditions for robustness, stability, and performance with respect to unmodeled plant dynamics and disturbances.

(2) Regions and rates of convergence with respect to unmodeled dynamics and disturbances.

These results extend the theory of adaptive control to more practical engineering environments. They also provide the means to monitor the stability, performance, and convergence of adaptive systems during transients. Thus, the theory establishes engineering guidelines for an outer control loop whose purpose is to coordinate system performance and to provide intelligence for changing plant parameters and operating conditions. These results are also directly applicable to nonlinear and multi-rate control laws for nonlinear systems.

The results produced in Phase I are the product of several years of collaborative efforts to develop a theory of robustness for adaptive systems. A summary of the Phase I research is contained in Section 2. Related selected references include Kosut and Friedlander (198?, 1985), Kosut and Anderson (1984, 1985) and Kosut and Johnson (1984, see also Appendix A).

1.3 SUMMARY OF PREVIOUS WORK AT ISI - REAL-TIME IMPLEMENTATION

Concurrent with the Phase I theory development, ISI has also pursued the development of a real-time control processor which is capable of implementing nonlinear control laws, including parameter adaptive control, on-line system identification, and gain-scheduled control. This work is currently being supported in part by internal funding and by the U.S. Army (AMCCOM). Past support has also come from the U.S. Air Force (AFWAL). The real-time processor utilizes the same data structures as ISI's existing simulation and analysis software product MATRIX_X (see Section 3). Hence, candidate designs can be rapidly implemented and tested in a laboratorv environment.

ISI's processor is named the MAX 100 (<u>MA</u>TRIX$_X$ <u>A</u>rchitecture E<u>x</u>ecutive). The MAX-100 will be an essential element for testing prototype control laws and provides:

(1) rapid implementation of control laws - no real-time code need be written;

(2) extensive diagnostics for debugging;

(3) easy maintenance - MAX-100 shares model catalogs and data bases with MATRIX_X.

A detailed discussion of these issues can be found in a paper by Shah, Walker, and Saberi (1985) included as Appendix B. Other references include Walker, Shah, and Gupta (1984) and Shah, Floyd, and Lehman (1985).

Though the development of this processor has provided significant experience in real-time control systems, further work is needed to develop a complete system for aerospace use.

1.4 SCOPE OF REPORT

The next section is orgaanized to emphasize the parallel development of the software simulation/analysis toolkit and the compatible architecture and building of a real-time processor. The theory is reviewed followed by a description of required hardware architecture.

Section 3 contains an overview of related work. Basic references and appendices on adaptive control complete this report.

SECTION 2

RESULTS OF PHASE 1 RESEARCH

This section provides a detailed summary of Phase I results. The Section is divided into a discussion first of software tools for adaptive and nonlinear algorithms, and then a discussion of real-time processor implementation issues.

2.1 SOFTWARE TOOLS FOR ADAPTIVE SYSTEMS

Successful implementation of adaptive systems requires close attention to theoretical algorithmic aspects and hardware/software compatibility. Adaptive control schemes have the generic form shown in Figure 2-1 and differ only in internal characteristics such as:

(1) model parametrizations
(2) parameter adaptive algorithms
(3) control design algorithms

The basic software tools which are needed for analysis of adaptive systems include:

(1) Interactive graphical block diagram manipulations.

(2) Analysis tools for evaluating system performance, e.g., stability and parameter convergence.

The $MATRIX_X$ software system currently incorporates the SYSTEM_BUILD feature (see Section 3) which is capable of general block diagram manipulation, but will have to be specialized further for adaptive system forms.

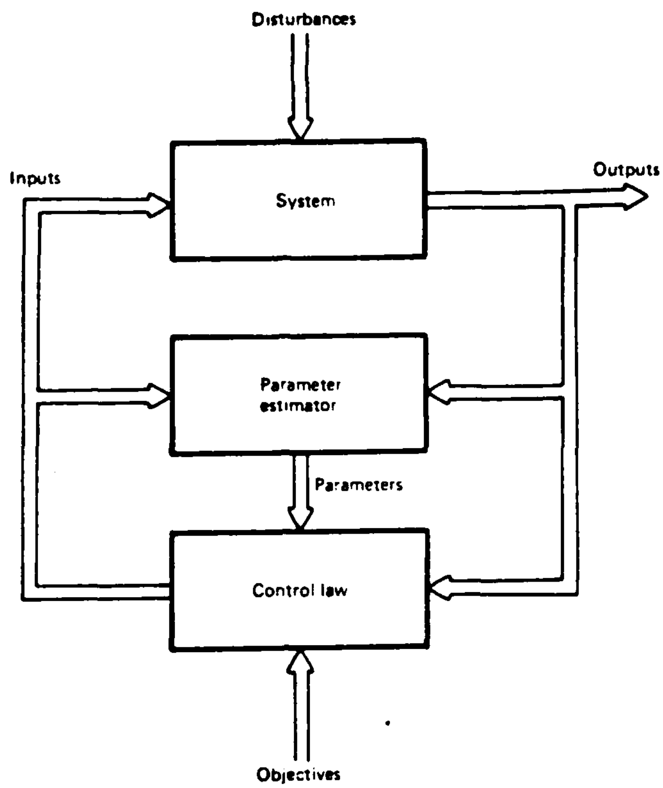


Figure 2-1. Adaptive Control System

Specifically, the software tools should allow the user to select from a catalog of standard adaptive system algorithms corresponding to the internal characteristics listed above. For example, the user could select from the following types of catalogs:

Model	Control Design	Adaptation
ARMAX	Model Reference	Gradient
State-Space	Self-Tuning	Recursive Least Squares
	Pole-Placement	Recursive Max Likelihood
		Extended Kalman Filter

Within each sub-topic the user would then be given a more detailed choice based on more detailed characteristics and a priori system knowledge of dynamics and disturbances. Of course, the experienced user can by-pass those choices and create any other algorithm.

The tools needed for evaluating performance do not depend on the user choice. These tools include testing operators for SPR (strict positive real), and testing signals for persistant excitation. forming the ODE (Ordinary Differential Equation), and performing averaging analyses. The following subsections briefly describe these tests.

2.1.1 SPR Condition and Testing

The SPR condition arises in the proof of stability of practically all adaptive schemes, e.g. Kosut and Johnson (1984) and Appendix A. Hence, it is very important to provide the means to test that the SPR condition holds. This will involve both on-line and off-line testing proceedures. For example, in analyzing a particular combination of adaptive algorithm and system model, the SPR test is off-line. In the case where the algorithm is being implemented in the real-time processor, the test is on-line.

To illustrate how the SPR condition arises, consider a system in ARMAX form,

$$A(q^{-1})y_t = B(q^{-1})u_t + C(q^{-1})v_t$$

where A, B, and C are polynominals in the delay operator q^{-1}, i.e., $q^{-1}x_t = x_{t-1}$. A typical identification model of the above system has the linear predictor form

$$y_t = \theta^T\phi_{t-1}$$

$$\phi_{t-1} = [y_{t-1}, y_{t-2}, \ldots, u_{t-1}, u_{t-2} \ldots]^T$$

However, the actual (optimal stationary) predictor is

$$y_t = \frac{1}{C(q^{-1})} \quad \Theta_o^T \phi_{t-1}$$

where θ_o consists of the coefficients in $A(q^{-1})$ and $B(q^{-1})$. As a result of this approximation the SPR condition arises to prove stability, e.g., $[C(q^{-1}) - \frac{1}{2}]$ should be SPR for the least squares update and $C(q^{-1})$ should be SPR for the gradient update (see e.g., Ljung and Soderstrom, 1983, or Goodwin and Sin, 1984).

By stability is meant that the parameter estimates and all relevant signals are bounded. The parameter estimates may not converge to the true value nor to any fixed value. Conditions for parameter convergence involve a richness in frequency content of the regressor vector $\phi(t)$, refered to as persistent excitation, which is discussed below.

A multivariable transfer matrix $H(q^{-1})$ can be tested for SPR in the frequency domain (Desoer and Vidyasagar, (1975). The result is that $H(q^{-1})$ is SPR if and only if the transfer matrix

$$G(q^{-1}) = (I + H(q^{-1}))^{-1}$$

is stable and

$$\sigma_{max}\, [G(e^{j\theta})] < 1, \quad -\pi<\theta<\pi$$

where σ_{max} denotes the maximum singular value. Once having obtained $H(q^{-1})$ it is a straightforward matter to perform the singular value test using existing $MATRIX_X$ software. The difficult part is to obtain $H(q^{-1})$ in the first place, for a given adaptive system. The requisite $H(q^{-1})$ depends on the algorithm and tools needed to obtain it both on line and off-line, (see Kosut and Johnson, 1984 in Appendix A for a precise definition of $H(q^{-1})$).

2.1.2 Persistent Excitation

When the appropriate operator $H(q^{-1})$ is SPR, the adaptive system is stable and, hence, the adaptive parameters are bounded. To guarantee parameter convergence requires that the regressor vector $\phi(t)$ is PE (persistently exciting), i.e., there exists an integer N and a positive constant α such that

$$\lambda_{min} \left\{ \frac{1}{N} \sum_{t=s}^{s+N-1} \phi(t)\phi(t)^T \right\} \geq \alpha$$

for all integers s, where λ_{min} is the minimum eigenvalue (see e.g. Anderson and Johnson, 1982). The ability to perform the PE test is a necessary tool for on-line and off-line monitoring of adaptive system performance.

2.1 3 Tools for ODE Analysis

The ODE (Ordinary Differential Equation) analysis for adaptive systems, developed by Ljung (1977), is used to determine the adaptive system asymptotic behavior. The basic idea is that the parameter estimates $\hat{\theta}(t)$ of the adaptive system:

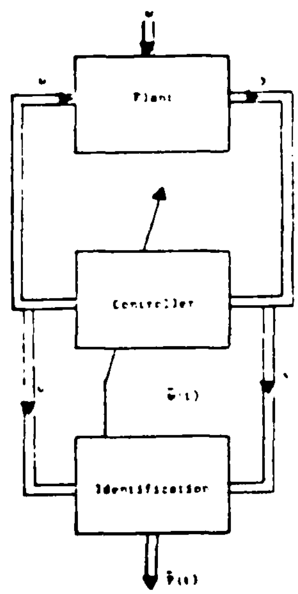


Figure 2-2. Basic Structure for ODE Analysis

will asymptotically approach the trajectories of the ODE:

$$\frac{d}{dt}\bar{\Theta} = R^{-1}f(\bar{\Theta})$$

$$\frac{d}{dt}R = G(\bar{\Theta}) - R$$

The nonlinear functions F and G can be estimated using the covariance estimate $\hat{P}(t)$ and the known plant/controller combination (see Ljung, 1977 for details).

The ODE analysis is based on a theory of stochastic averaging and assumes that the parameters are near convergence. Hence, the results provide necessary conditions for convergence. More general averaging results are described next. These results, in some cases, are necessary and sufficient for parameter convergence.

2.1.4 Tools for Averaging

Adaptive algorithms generally have the form

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) + \varepsilon\, f(t-1,\hat{\Theta}(t-1),\xi(t-1))$$

$$\xi(t) = g(t-1,\hat{\Theta}(t-1),\xi(t-1))$$

where $\hat{\Theta}(t)$ is the adaptive parameter estimate and $\xi(t)$ is a vector consisting of all other system states. The functions f and g correspond to the algorithm and system description respectively. The parameter ε is the step-size. When ε is small we can analyze the "average" system

$$\hat{\theta}(t) = \hat{\Theta}(t-1) + \varepsilon\, \bar{f}(\hat{\theta}(t-1))$$

where $\bar{f}(\Theta) = \frac{1}{N} \sum_{t=s}^{s+n-1} f(t,\Theta,o)$

Thus $\hat{\Theta}(t)$ is "slow" compared to the "fast" states $\xi(t)$. In the case when the system is linear the convergence properties of the averaged system can be determined in the frequency domain using a generalized positivity test of the form,

$$\lambda_{min} \left\{ \sum_{m=o}^{\infty} \operatorname{Re}[\alpha_m \alpha_m^*] \operatorname{Re} [H(e^{j\omega_m})] \right\} > 0$$

where α_m and ω_m are, respectively, the Fourier series coefficients and exponents of the regressor vector, and $H(q^{-1})$ is the operator as defined in Section 2.1.1 (see e.g. Riedle and Kokotovic, 1984; Kosut, Anderson, and Mareels, 1985).

This condition is significantly weaker than the usual SPR condition $\operatorname{Re}(He^{j\omega}) > 0$ which is tested for all frequencies ω. The above averaging test is a sum and $\operatorname{Re} H(e^{j\omega}) > 0$ is required only when $\operatorname{Re} [\alpha_m \alpha_m^*]$ is large.

The class of signals which can be tested in this way is larger than those allowed in the ODE theory. Also, the test is easily implemented by using the standard signal analysis techniques to find the Fourier series of a function. One immediate result is that if the test fails it is possible to augment the input signals in the frequency range where more signal is needed. Those features are easily incorporated into the algorithm as well as the real-time system.

2.2 NONLINEAR CONTROL DESIGN BY GAIN SCHEDULING

Adaptive control can be considered as a nonlinear control with a special structure. A highly successful approach to nonlinear control design which shares this same special structure, is gain-scheduling. In fact, this approach can be applied to any nonlinear control problem. Because parameter adaptive control and gain scheduling control share the same basic structure, the real-time processor and software/simulation toolkit can easily provide for both. The following discussion illustrates the basic requirements for gain scheduling

A typical gain-scheduled control system is shown in Figure 2-3, where P is the plant (aircraft), d is an output disturbance, and C is the gain-scheduled controller, whose gains are scheduled as a function of the actual trajectories (u,y). Note that C maps output error signals $\tilde{y} := \bar{y} - y$ into control error signals $\tilde{u} = u - \bar{u}$.

The first step in obtaining C is to design a collection of linear controllers based on linear models, each corresponding to a particular reference trajectory $(\bar{u},\bar{y})$. Thus, the resulting controller gains are a function of the reference condition $(\bar{u},\bar{y})$. The control gains are then "scheduled" as functions of the actual condition (u,y) to achieve a <u>continuous</u> <u>nonlinear</u> control throughout the operating envelope.

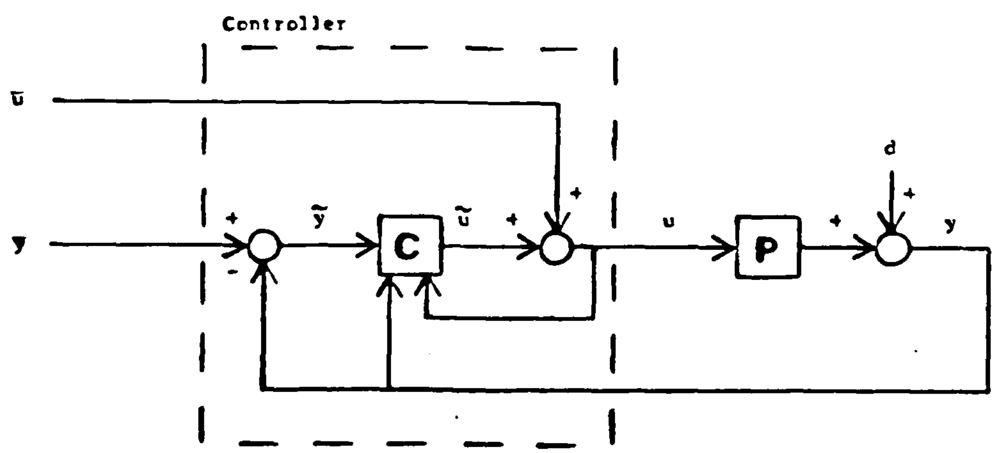


Figure 2-3. Gain-Scheduled Control System

Scheduling the gains can be accomplished using a variety of interpolation or regression fitting schemes. The choice depends largely on the capabilities of the hardware in which the algorithm will be implemented and the complexity of the gain schedule. This technique is well established, having been applied to many flight control systems.

The gain scheduling design procedure involves several models of the actual plant P. A high fidelity nonlinear model, denoted by P_{NL} is developed either analytically from physical laws, or numerically from flight data. Often, P_{NL}, is a combination of both. A set of reference trajectories $(\bar{u}_i, \bar{y}_i)$, $i = 1, \ldots, k$ are determined from P_{NL}, i.e.,

$$\bar{y}_i = P_{NL}\bar{u}_i, \quad i = 1, \ldots, k$$

Each reference $(\bar{u}_i, \bar{y}_i)$ generates a <u>linear</u> <u>perturbation</u> <u>model</u>, denoted by P_{L_i}. This model can be obtained as a first order perturbation of P_{NL}, i.e.,

$$P_{NL}(\bar{u}_i + u) = P_{NL}\bar{u}_i + P_{L_i}\tilde{u} + 0(||\tilde{u}||^2)$$

Repeating for k selected flight conditions $(\bar{u}_i, \bar{y}_i)$, $i = 1, \ldots, k$ yields a corresponding collection of linear models $\{P_{L_1}, \ldots, P_{L_k}\}$. In the case when $(\bar{u}_i, \bar{y}_i)$ is an <u>equilibrium</u> then $\bar{u}_i$ and $\bar{y}_i$ are constants and P_{L_i} is LTI. When $(\bar{u}_i, \bar{y}_i)$ is a <u>dynamic</u> <u>trajectory</u> then P_{L_i} is LTV (linear time varying). Usually only equilibria are selected, however, to fully account for the behavior of high performance aircraft, it is necessary to consider dynamic references as well as equilibrium references.

Having determined a collection of linear models $\{P_{L_1},\ldots,P_{L_k}\}$ corresponding to the k-nominal trajectories $\{(\bar{u}_1,\bar{y}_1),\ldots(\bar{u}_k,\bar{y}_k)\}$, any number of design techniques can be used to determine a set of linear controllers $\{C_{L_1},\ldots C_{L_k}\}$. The ith linear controller C_{L_i} is indirectly a function of the ith trajectory $(\bar{u}_i,\bar{y}_i)$. Connecting this collection of controllers as a function of the actual (u,y) is "gain scheduling". The resulting controller C is nonlinear. The same scheduling procedure can be used to connect the collection of linear models. The resulting nonlinear model is often referred to as a "simplified nonlinear model", denoted by P_{SNL}.

A fundamental issue in the gain-scheduling procedure is that there is no theoretical justification that the resulting nonlinear (gain scheduled) control will provide acceptable performance while the vehicle is in transit from one flight/power condition to another. The difficulty lies in the fact that the linear models are only known to be valid at specific conditions. The effectiveness of the linear model P_{L_i} in the neighborhood of the ith reference $(\bar{u}_i,\bar{y}_i)$ can be evaluated from the test set-up shown in Figure 2-4.

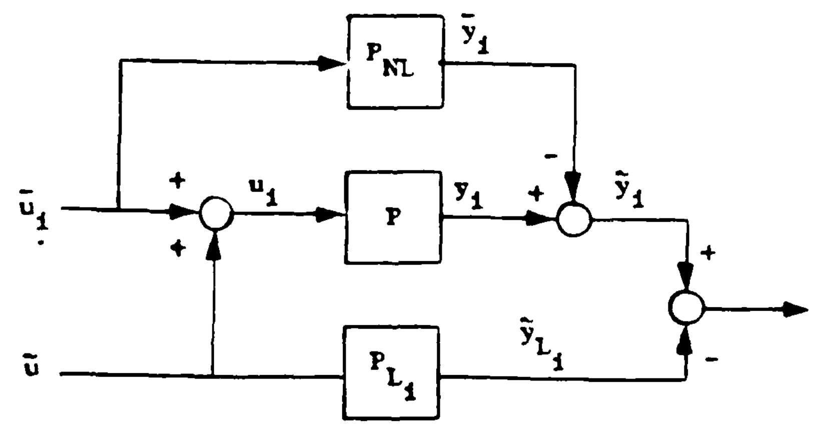


Figure 2-4. Model Error Set-Up

Details on this nonlinear model error testing procedure can be found in Appendix C.

2.3 SYSTEM ARCHITECTURE

<u>Multi-rate Control Laws on Multiple Processors</u>

Multi-rate control laws refer to hierarchical control laws consisting of systems with more than one sample rate. In the context of adaptive control it is typical to perform identification update at a slower rate than the control law implementation.

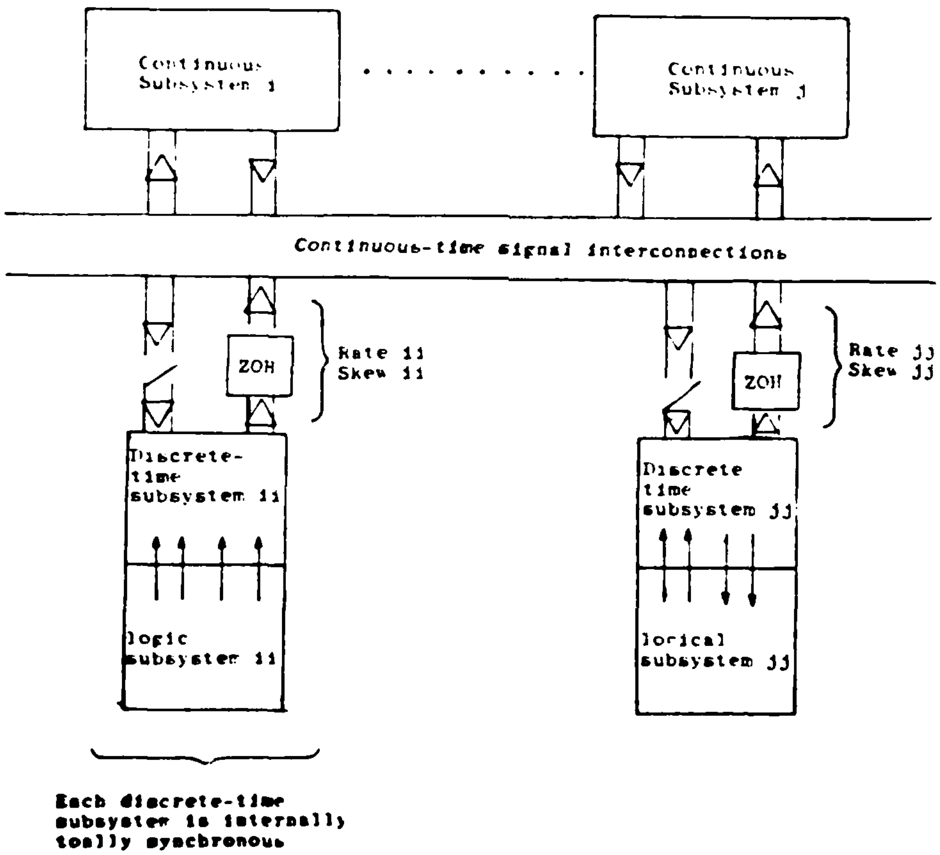


Figure 2-5. Basic Multi-Rate Control System

Note that each of the subsystems has a synchronous timing and can typically be considered as performing a separate task. This decomposition corresponds to the user-defined decomposition based on the functionalities of the control law subsystem. Often these functionalities are defined by appealing to separation of time scales.

Some of these tasks can be tied to a specific processor in the system. In particular, D/A, A/D, and continuous signal prefiltering functions are typically tied to a processor capable of performing analog I/O. Those tasks that run on the same processor have to share the computation resources.

Strategies for scheduling these tasks, assigning them to specific processors, managing synchronization, and sharing of data among tasks running on many processors requires a careful design. Implementation of such a system requires significant effort in systems programming. Note that each of the tasks have to be executed periodically. Failure to finish any of the tasks on time constitutes a potential failure of the overall control law.

In order to ensure integrity of timing. a real-time clock must provide interrupts for scheduling all periodic activities in the system.

Hardware Subsystems

Typical algorithms used for adaptive control and gain scheduled control impose computational burdens that require special purpose hardware for implementation if the parameter update rates exceed 20Hz on systems of order ten or greater (see Shah, Walker, and Saberi, Appendix B).

Among the various functions required by such a hardware system are:

I. Analog to Digital to Analog conversion, Antialiasing filters with programmable bandwidth. Accurately timed A/D and D/A operations to provide clean data.

II. High performance (2MFLOPS or more) floating pt. processor capable of performing the entire nonlinear block diagram update without imposing system bus traffic overhead. Tasks that could be performed in parallel should be split and loaded in individual floating point processors thereby increasing the overall throughput.

III. User interface, overall system management, disk I/O management

It is natural to specify an architecture where individual modules are specialized to handle each of the three tasks above.

Major vendors such as Intel, DEC, and Motorola provide modules which provide general purpose machines in the form of single-board computers, disk controllers, CRT/video controllers, etc.

Analog I/O Module

There are numerous vendors that supply A/D, D/A cards for various bases but most of them are specialized for Data-Acquisition rather than real-time control so that clean data with simultaneous sampling of channels at high throughput rates is not easily achievable. I/O subsystems with integral programmable antialiasing filters, accurate timing and low system overhead are not available and have to be designed. Such a module should also include autoranging capability for an increased numerical range (up to 20 bits). The throughput rate per channel must be at least 5KHz. The analog I/O module must work without needing servicing from other processors.

Floating Point Module

High throughput numerical modules available from vendors such as SKY computers can provide about 1MFLOPOS for vector operations. In order to achieve 5 to 10 MFLOP effective throughput a floating-point board specialized for numerical functions involved in adaptive control and nonlinear control should be designed. Such a processor would have the following basic architecture.

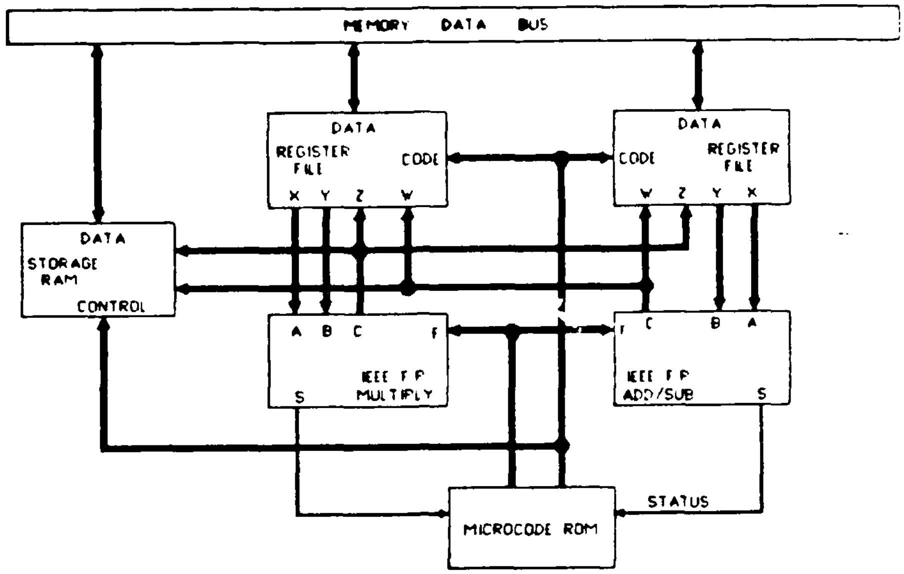


Figure 2-6. Structure of the Simulation Engine

The floating point simulation engine consists of a fast multiplier and a fast floating point ALSL for arithmetic operations such as add, subtract, absolute value, fix-float conversions, normalization, and comparisons. The register files above act as a high speed pipelined multipost switches capable of routing both data and code at the speed of the two processing elements (PE). The microcode ROM sequences orchestrates the activities of the PEs and the register file switches. Local storage of data and instructions is on a high speed RAM on the left.

The key is to have all the computations required for function evaluation in real-time control block diagram update performed on the high speed floating point module with no intermediate system bus transfers.

This goal requires development of the following capabilities:

(1) Basic arithmetic operations +,-,*, divide, absolute comparisons.

(2) Square root, trignometric, and hypertrignometric functions, log and exponential functions.

(3) Vector operations.

(4) Individual block updates consisting of vector operations

$$x_{k+1} = F(x_k, u_k)$$

$$y_k = H(x_k, u_k)$$

(5) Individual subystem update consisting of block updates and resolution of interconnections among blocks.

System Management and General Purpose Machines

Among the available microcomputer families and buses, Intel's Multibus based single-board computers provide the best match for high performance and functionality offered by their product line. An important criterion in selecting Intel's family of products is the availability of hardware floating point modules 80287 and 8087 along with a mature set of languages and operating tools to handle the entire family of products.

Figure 2-7 describes one particular configuration currently being used for a real-time control application.

2.4 REQUIREMENTS FOR SOFTWARE AND HARDWARE

A representative toolkit for various phases of adaptive and nonlinear control mechanization can now be described. There are two distinct environments - an analysis and simulation environment and a real-time implementation environment. Both environments share the same model definition data structure. The top three blocks in Figure 2-8 represent analysis/simulation, while the bottom two refer to real-time processing.

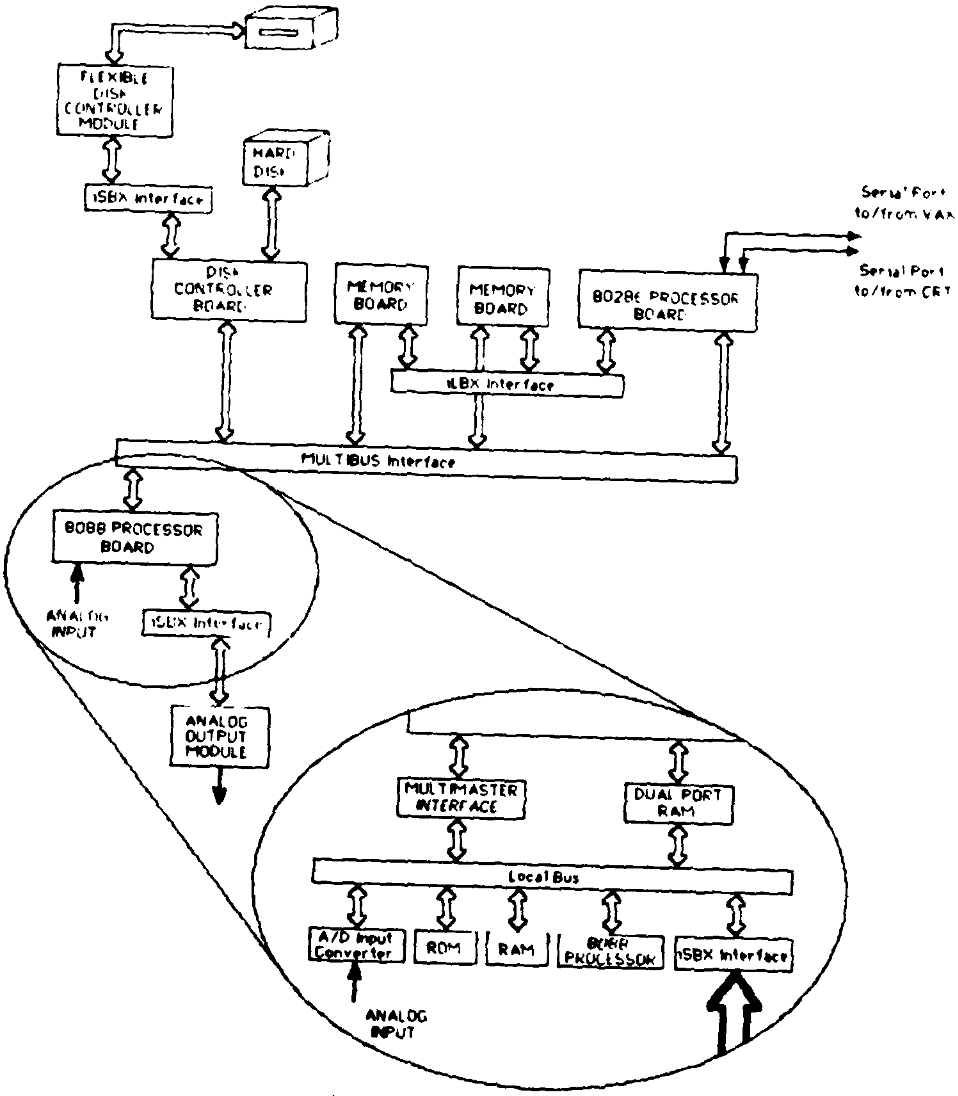


Figure 2-7. Typical Real-Time Control Hardware Configuration

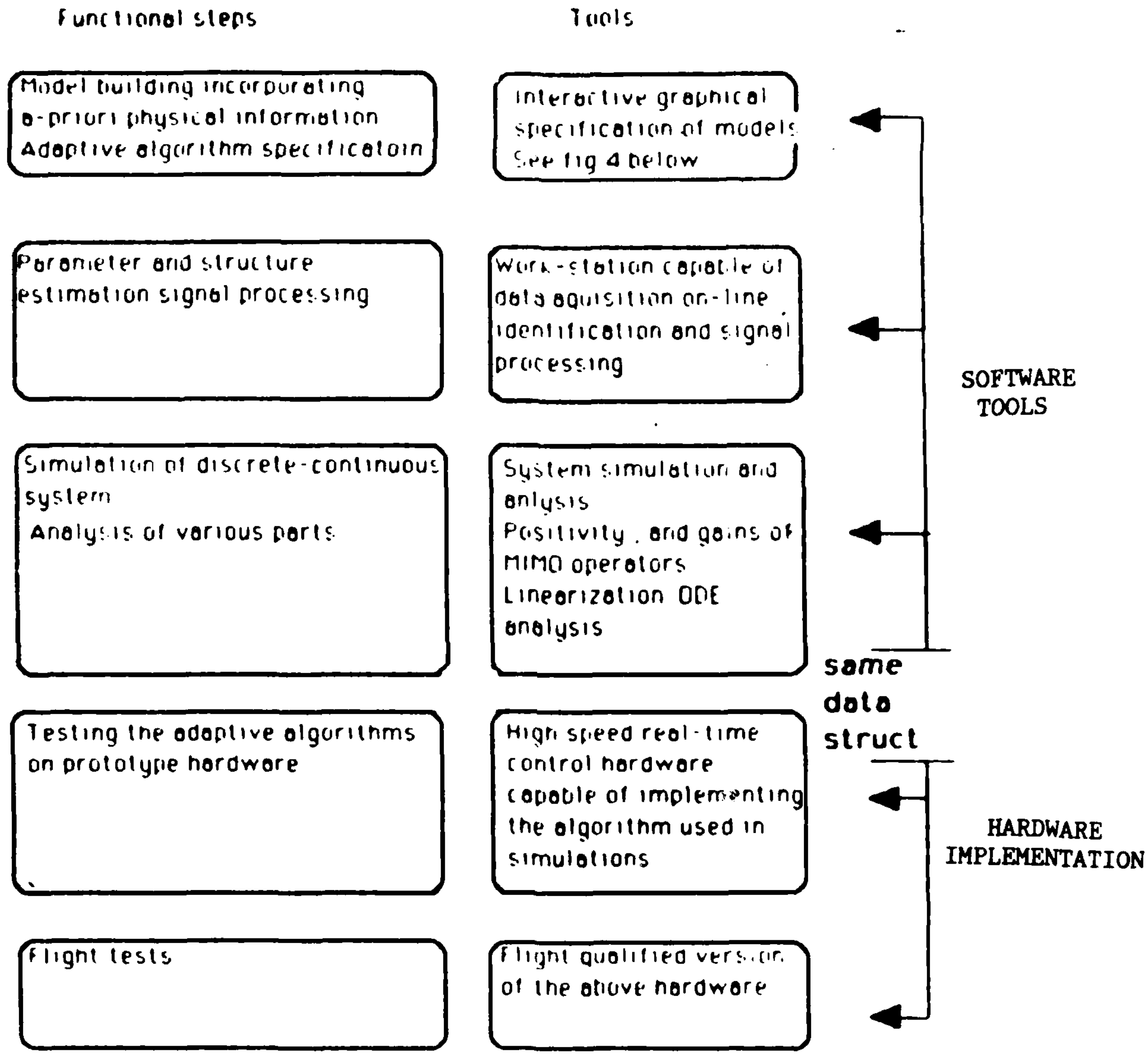


Figure 2-8. Adaptive Control Mechanization Process and Tools

The needs for computer-aided development and implementation of adaptive and nonlinear control laws are:

(1) Development of a user-friendly software simulation and analysis toolkit for various phases of adaptive and nonlinear control mechanization.

(2) Development of an efficient transfer procedure from the simulation/analysis environment to the real-time processor environment.

2.4.1 Software Simulation/Analysis Toolkit

Specific objectives to develop the toolkit are as follows:

I. Provide an interactive graphical model building tool with the following features:

   (1) Dynamical systems of nonlinear differential equations can be developed from block diagrams.

   (2) The models can have a hierarchical structure - submodels can be imbedded inside larger models.

   (3) Model catalogs can be treated like data bases.

   (4) Nonlinear models can be linearized about selected equilibriums or nominal trajectories.

   (5) Model error measures which are used in robustness testing can be easily obtained for either linear on nonlinear systems.

II. Provide a work-station capable of data acquisition, on-line system identification and signal processing with the following features:

   (1) Parametric models for system identification can be developed from a catalog of choices or from block diagram building.

   (2) Data for system identification can be acquired either from a simulated evaluation model (as in I) or from a real-time data source.

III. Provide an interactive adaptive control and identification algorithm building tool such that:

(1) The novice user can rely on automatically selected algorithms based on the structure of the problem. Default parameters for the algorithm should work in most cases.

(2) The expert user can select the algorithm and the associated design parameters to gain full control over the optimization.

(3) Extensive diagnostics can be made available in case of algorithmic difficulty, ill-conditioning of the problem or the algorithm. Heuristics to overcome these difficulties should be programmed where appropriate. The user should be able to obtain explanation of the heuristics and choices of alternatives Some of the diagnostic tools include on-line and off-line testing procedures for:

(a) passivity and other input/output sector conditions
(b) persistent excitation of signals
(c) stability and performance robustness
(d) parameter convergence
(e) linearization
(f) ODE and averaging analyses.

The results of the Phase I effort have provided the basic theory and means to perform many of these diagnostic tests

2.4.2 Design of the Hardware - Real-Time Control System

The real-time control system design objectives are as follows:

I. Provide a simple system for rapid implementation of adaptive and nonlinear real-time control systems which facilitates the prototype development and testing phase of control law implementation. It should operate on the data-base generated by the software analysis/ simulation tool-kit.

(1) The real-time control interface for editing and examining should be in terms of the graphical block diagram entered by the user in the design and simulation software.

(2) The user should be able to modify various aspects of this block diagram on the real-time control system so that the real-time system can be operated without requiring access to the design and simulation software.

(3) The communication link between the design software and the real-time control system must be a fast and single command on each system.

II. The real-time system should be able to process analog signals from a physical plant, provide performance measures of the controlled system, and perform data acquisition.

(1) The user should be able to connect the real-time control system directly to the physical plant being controlled with standardized analog signals (4-20 ma current loop or ±5v bipolar.

(2) The sampling of analog signals must be precise and none of the sample times should be dependent upon the computation delays.

(3) The real-time control system must have extensive error handling capabilities that relate to computational burden, numerical exceptions, missed timings, and integrity of shared data in a multiprocessor environment. Performance measures include signal statistics, idle time on various processors, and memory usage. The user should be able to configure various statistical algorithms to operate on top of the control system to give performance data and store it on a secondary storage such as a hard disk.

III. Provide high computational capability as well as numerical accuracy as required during the prototyping phase.

(1) Provide both floating point and fixed point arithmetic. Floating point arithmetic format should conform to the IEEE 754 standard. Most high performance floating point hardware being designed today conforms to the IEEE floating point format.

(2) Prototyping phase requires the user to experiment with options that usually require more computational capabilities than the one required by the target system hardware. A floating point hardware capability in excess of 5 million floating point operations per second is required for achieving 10-100 hz identification update rate on systems of order 5 to 10.

(3) The architecture of such a floating point hardware must be optimized to give high throughput for real-time control algorithms. In contrast with array processor architectures available today, the pipelining of data and instruction must not be allowed to generate loop delays.

IV. It should require no real-time programming from the user. The user-interface should make the system readily accesible to non-programmers.

All the considerations described for the user interface above apply to the user interface of the real-time control system.

Section 3
RELATED WORK

Integrated Systems, Inc. (ISI) has been performing a significant amount of work in the general area of computer-aided-engineering for system analysis with particular emphasis on control design and simulation. The work involving development, maintenance, and enhancements to MATRIX_x was funded by ISI internally.

ISI has also developed a real-time control design processor under U.S Army funding. The processor has been modified for general purpose use and is called the MAX-100. Several projects are researching other aspects of CAE technology for control design, implementation, and validation. MATRIX_x currently is being used by over thirty companies, research laboratories, and universities, including Lockheed, General Motors, M.I.T., and the Air Force.

3.1 IN-HOUSE DEVELOPMENTS: MATRIX_x

MATRIX_x provides the following interactive capabilities:

1. Control and estimator design
2. System identification and signal processing
3. Interactive model building (SYSTEM_BUILD)
4. Simulation and evaluation

Matrix algebra, interactive graphics, and model catalog management support these capabilities. Figure 3-1 shows the structure of MATRIX_x. In the following, we describe the control design, model building, and simulation capabilities of MATRIX_x.

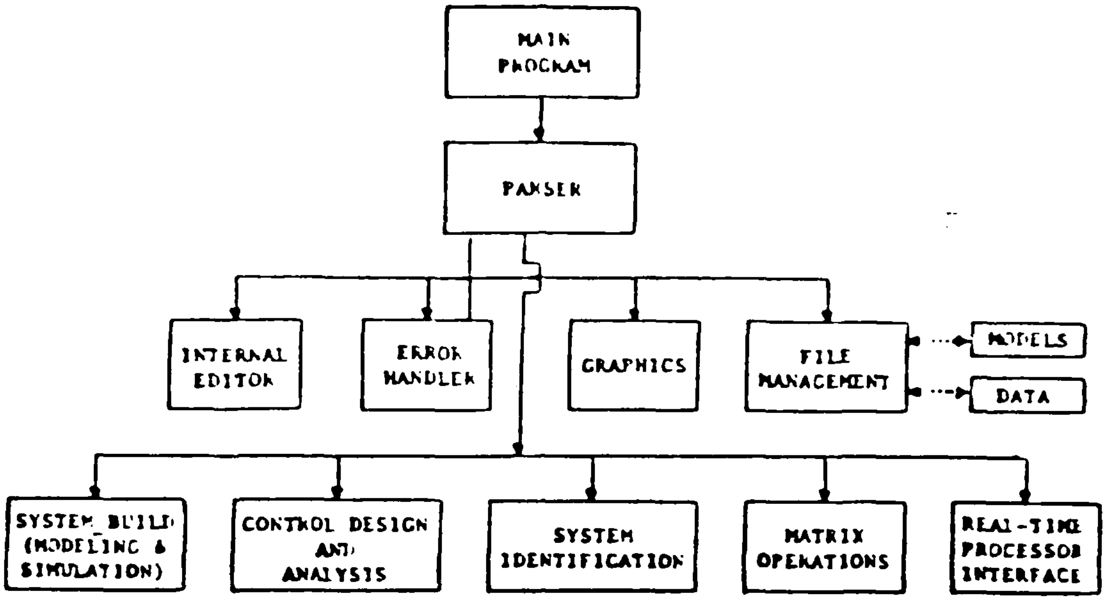


Figure 3-1. MATRIX$_X$ Software Architecture

Control Design and Analysis

Control design in MATRIX$_X$ can be based on any of the following:

(a) Classical methods including root locus, Bode, Nyquist, and Nichols (single-input/output or multivariable plants)

(b) Linear-Quadratic-Gaussian (LQG)

(c) Methods based on A-B invariant subspaces

(d) Eigenstructure assignment and zero placement

(e) Adaptive control using self-tuning regulators and other techniques

These capabilities are available for use with both continuous and discrete systems.

For the LQG problem, the algebraic Riccati equation is solved from extended Hamilton equations, avoiding inverses which are troublesome in the singular case. The equations are row compressed with an orthogonal transformation followed by the QZ pencil decomposition.

TABLE 3-1. MATRIX$_X$ CAPABILITIES: CONTROL DESIGN AND SYSTEM ANALYSIS CAPABILITIES (APPLICABLE TO CONTINUOUS, DISCRETE AND HYBRID SYSTEMS)

<u>Classical Tools</u>
Root Locus Bode Plots Nyquist Plots Nichols Plots
<u>Modern Tools</u>
Optimal Control Design, Discrete and Continuous Optimal Filter Design, Discrete and Continuous Frequency-Shaped LQG Design Singular-Value Decomposition of the Return Difference Eigensystem Decompositions Including the Jordan Canonical Form Model Following Control Model Reduction Linearization of Nonlinear Systems Minimal Realization and Kalman Decomposition Geometric Control Algorithms Multivariable Nyquist Plots

Extensions to LQG methods require inclusion of the dynamics of the reference inputs, disturbances, sensors, and actuators. Appending dynamics in frequency-shaped control design or model-following techniques involves forming augmented equations This is easily accomplished with MATRIX$_X$ primitives. Use of frequency-shaped cost functionals, with singular value plots for robustness evaluation, allow incorporation of engineering judgment in control design.

Evaluation tools for linear systems include frequency response and power spectral density plots, time responses, and determination of transmission zeros. The principal vector algorithm (PVA) primitive for numerically reliable extraction of the Jordan Form (with discriminatory rank deflation of root clusters) us useful in modal analysis. PVA enables computation of residues or partial fraction expansions of multivariable systems. Some of these capabilities are listed in Table 3-1.

$MATRIX_X$ also provides capabilities to transform models from any one of the following forms to another:

(i) Discrete or sampled data

(ii) Continuous

(iii) State space including various canonical forms

(iv) Transfer functions

(v) Poles, zeros, and gains.

Interactive Model Building (SYSTEM BUILD)

The interactive model building facility called SYSTEM_BUILD is a tool for building models of complex systems for use in simulation, control design, and trade-off studies. The user can develop multi-input/multi-output (MIMO) system models from models of individual parts of the system. Transfer function descriptions can be combined with nonlinear functions and state-space models. It is also possible to connect an externally defined FORTRAN module to models defined in SYSTEM_BUILD. Models can be placed in catalogs for future use. Systems defined using SYSTEM_BUILD can be linearized and simulated with arbitrary inputs. Modules or parts can be changed or replaced without recompiling and relinking FORTRAN code.

A hierarchical structure allows models to be developed "top-down" or "bottom-up." In the top-down approach, the designer specifies an overall system in terms of its major subsystems. Each major subsystem can be defined as an interacting interconnection of lower level subsystems. The lowest level subsystems are finally specified using basic elements, which might consist of nonlinearities, table look-ups, transfer functions, state-space models, and summing junctions. Nonlinearities can include saturation, absolute values, hysteresis, general piecewise linear functions-quantization and general algebraic nonlinearities. Transfer functions can be written as numerator/denominator polynomial coefficients, zeros/poles, or natural frequencies and damping ratios.

In the bottom-up building approach, the lowest subsystem models are developed first. Major subsystem and complete system models may then be assembled from lower-level system models.

Figure 3-2 shows the example of an automobile cruise control model developed under SYSTEM_BUILD.

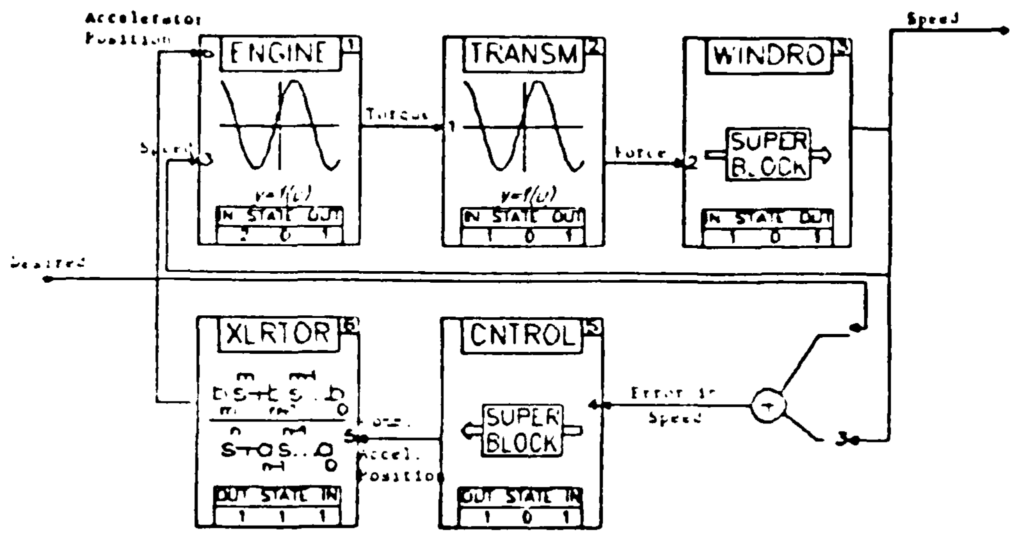


Figure 3-2. MATRIX$_x$ and SYSTEM_BUILD Block-Diagram Model of Cruise Control System for an Automobile (the figure shows actual screen)

Simulation and Analysis

$MATRIX_x$ provides capabilities for efficient linear and nonlinear simulation. Linear simulation is performed using a discrete representation and is structured to fully use sparseness in system matrices.

Table 3-2 shows the classes of systems treated within $MATRIX_x$. Table 3-3 shows the simulation and analysis capabilities.

A variety of integration algorithms is available for dealing with various classes of simulation models (Table 3-4).

TABLE 3-2. CLASSES OF SYSTEMS COVERED BY SYSTEM_BUILD

Linear
Nonlinear
Continuous
Single-Rate Discrete
Multi-Rate Discrete
Single-Rate Hybrid
Multi-Rate Hybrid

TABLE 3-3. SIMULATION AND ANALYSIS CAPABILITIES

Simulate general nonlinear continuous models
Simulate general nonlinear discrete models
Simulate general nonlinear multi-rate models
Simulate general nonlinear hybrid multi-rate models
Simulate to show transitions due to sampling
Study sampling and intersample behavior of hybrid and multi-rate system
Linearize nonlinear continuous models about initial conditions
Linearize nonlinear continuous models at a point along the trajectory
Linearize nonlinear discrete models about initial conditions (single sample rate)
Linearize nonlinear discrete models at a point along the trajectory

TABLE 3-4. INTEGRATION ALGORITHMS AVAILABLE IN $MATRIX_X$

Euler	- Euler
Rk2	- Runge-Kutta (2nd order)
Rk4	- Runge-Kutta (4th order)
Kutta-Merson (fixed step)	
Kutta-Merson (variable step)	
DASSL	- implicit stiff predictor-corrector

3.2 REAL-TIME PROCESSORS: MAX 100

ISI has developed a system called MAX-100 for testing control laws in the laboratory. MAX-100 uses a multiprocessor configuration to emulate nonlinear control laws. The real-time programming requirements are minimized by using off-line design system $MATRIX_X$ to specify the control laws. MAX 100 uses the Intel 80286/287 and 8086/87 processors.

The proposed prototype can use the experience gained in MAX-100 development directly. To gain high speed and flexibility, we are now proposing use of Multiplier ladder chips rather than the standard microprocessors used in MAX-100. Research and development will also make it possible to develop a high speed control processor board for use in flight testing and prototype system development.

3.2.1 MAX 100 Real Time Control and Data Acquisition System

Description: MAX-100 is a high performance real-time control implementation and data acquisition system. It provides the ability to perform nonlinear real-time control, on-line system identification and data acquisition as well as digital signal processing. It can be connected with ISI's control design and modeling software package, $MATRIX_X$, for control law design and downloading of control logic It also can be run as a standalone system for field use.

MAX-100 provides a vital link in the control design cycle for prototyping non-linear control laws with switching logic and adaptive control. A data acquisition facility is provided to debug control logic for off-line identification of model equations.

Equipped with two general purpose processors and two additional arithmetic floating point coprocessors, MAX-100 performs parallel computations for high real-time throughput. Baseline MAX 100 configurations handles 16 analog measurements while generating 8 analog control signals. Options allow expansions to 32 analog measurements and 16 control signals.

MAX-100 derives its name from $MATRIX_X$ Architecture Executive, because it implements $MATRIX_X$ run time library for real-time use.

FEATURES:

- Nonlinear block diagrams mapped into a multiprocessor implementation
- 32-bit IEEE-standard floating point arithmetic
- Graphical block diagram specification, editing, simulation and analysis in $MATRIX_X$ System_Build
- Push-button simplicity in implementation of control logic.
- Data acquisition and storage concurrently with real-time control.
- On-line identification with linear and nonlinear models.
- Adaptive control and fault detection capabilities.
- Multi-rate control implementation feature.

A High Productivity Tool

The most remarkable aspect of MAX 100 is the ease of use. For example, a graphical block diagram for a nonlinear control law (using the System_Build feature of ISI's $MATRIX_x$ design software) can be implemented directly on the MAX-100. Typically prior knowledge and physical laws are used to generate a mathematical model of the plant in System_Build. However, MAX-100 can perform data acquisition and on line identification to refine parameters of such a nonlinear model. Frequency domain methods, based on spectrum analysis, or time domain methods, based on parametric estimation methods, are available identification methodologies.

The refined model can be used for performing control synthesis and analysis in $MATRIX_x$. Once a satisfactory discrete-time control law is snythsized and simulated in $MATRIX_x$ with System_Build it can be implemented in MAX 100 with push-button simplicity. This control law can be nonlinear, have multiple sample rates and can be ultimately targeted to run on many processors.

MAX-100 handles such multi-rate control laws accurately. It handles sampling of signals and updating of outputs precisely in time without inserting any undersirable pipe-line delays or scheduling delays. It maintains synchronization and sharing of data among subsystems running at different rates without inserting any anaccounted delays which can deteriorate phase margins in control loops.

MAX-100 removes the burden of writing real-time code for control implementation and testing. It provides a highly reliable tool real-time control integrated with the design and simulation software.

Benefits

- A reliable tool for real-time control implementation prototyping and testing.
- An implementation tool integrated with design and simulation tools to provide a complete control design laboratory.

- Multirate architecture implemented as multiprocessor multi-tasking system for maximum throughput without inserting pipeline delays and attendant phase legs.

- Powerful control logic timing and debugging feature.

- Useful in the laboratory as well as in the field for control law timing and debugging.

- Common data base for real-time control implementation and off-line simulation to minimize potential for error.

3 3 FUNDED WORK

Five current projects are related to adaptive and nonlinear control implementation and associated control design CAE capability, discussed in this report.

Computer-Aided Design Methods for Engineering Analysis (National Science Foundation): The project is studying advanced numerical algorithms, hardware, software architectures to bring advanced mathematical research to a working engineer. A prototype workstation based on an IBM PC will be completed around March 1985

Automatic Real-Time Code Generation in Ada (Air Force Rocket Propulsion Laboratory): This project will lead to a capability for generating real-time Ada codes for control implementation once the control law has been specified using MATRIX_x. The code is written i Ada and will be portable to all imbedded processors with Ada compilers

Hardware for Control Implementation (AMCCOM): This project is developing a hardware system to allow automatic implementation of MATRIX_x designed control laws for real-time laboratory testing. The prototype will be ready in January 1985 and will be tested on a helicopter based gun turret.

Computer-Aided Design Methods for Optimization (Air Force Armament Laboratory): ISI is studying the feasibility of trajectory optimization for missile systems. This capability will be provided interactively in conjunction with the MATRIX_x SYSTEM_BUILD feature.

<u>Engineering Workstations for Distributed Parameter Systems (National Aeronautics and Space Administration, Langley Research Center)</u>: A study was done to develop distributed CAE workstations for modal testing and comparison of modal test data against finite element models.

Section 4

POTENTIAL APPLICATIONS

The adaptive and nonlinear control implementation hardware which would result from a successful Phase II would have major uses in commercial as well as Federal Government applications.

Commercial Applications

Advanced control methods are needed in chemical process control, robot control and design of fast robots, engine and suspension system control in automotive industry, and servo controllers in computer disc drive design. Many new applications will occur as improved CAE tools become available and the designed control laws can be rapidly verified in the laboratory. The proposed hardware will be capable of implementing optimal trajectories allowing more economical operation of process plants and automated manufacturing lines.

For example, the introduction of control design technology into the commercial environment has been slowed by the unavailability of easy to use Computer-Aided Control System Design (CACSD) tools Hardware implementations, refinements and tuning of selected control approaches are also hindered by the current need for real-time programming. Prior experience in paper mills and chemical plants has shown a two to three year time lag and 10-20 manyears of effort to implement and use a new control approach. The proposed products will reduce the development time of such systems to a few months.

Analysis using MATRIX_x has shown that paper mills could have saved $768,000 in just one year's fuel costs if adaptive and nonlinear control were continuously optimizing the use of recovery furnaces. Process control systems represent a $180 billion per year market. Most power plants control systems are over twenty years old and need to be replaced in this age of scarce resources.

Federal Systems

The principal results of developing a turnkey control implementation hardware will be lowered engineering costs, higher performance, and more rapid deployment of complex weapons systems new aircraft, complex transportation systems, and satellite communications systems. Advanced control, guidance and estimation methods can then be introduced into operational systems more rapidly since development times and risks will be reduced. Easy to use laboratory test tools may eventually change policymakers' decisions on appropriate funding levels for new complex systems development.

Advanced control will also lead to better weapon systems In many cases complex hardware will be replaced by simpler hardware coupled with advanced control. For example, the application of advanced guidance technology can extend certain missile performance by 100% without any extra fuel or hardware cost. New weapon systems like the advanced tactical fighter are so complex that stability, optimal performance and synchronization of various control features will be feasible only with advanced software.

Section 5

REFERENCES

Anderson, B.D.O., and C.R. Johnson, (1982), "Exponential Convergence of Adaptive Identification and Control Algorithms," Automatical, Vol. 18, No. 1, January 1982, pp. 1-13.

Dosoer, C.A., and M Vidyasagar (1975), Feedback Systems: Input-Output Properties, Academic Press, New York, 1975.

Goodwin, G., and K.S. Sin (1984), Adaptive Filtering, Prediction and Control, Prentice Hall.

Kosut, R.L., and B. Friedlander (1982), "Performance Robustness Properties of Adaptive Control Systems," Proc. 21st IEEE CDC, Orlando, FL, December 1982, pp. 18-23.

Kosut, R.L. and B. Friedlander (1985), "Robust Adaptive Control· Conditions for Global Stability," IEEE Trans. on Aut. Contr., to appear, July 1985.

Kosut, R.L., and B.D.O. Anderson (1984, 1985), "Robust Adaptive Control: Conditions for Local Stability," 234d IEEE CDC, Las Vegas, Nevada, December 1984; IEEE Trans. Aut. Control, to appear.

Kosut, R.L., and C.R. Johnson, Jr. (1984), "An Input-Output View of Robustness in Adaptive Control," Automatica, Special Issue on Adaptive Control, Sept. 1984, Vol. 20, No. 5, pp. 569-582.

Kosut, R.L., B.D O. Anderson, and I. Mareels, "Stability of Adaptive Systems: Methods of Averaging and Persistent Excitation", Proc. 24th IEEE CDC, to appear, Ft. Lauderdale, FL, December 1985.

Ljung. L. and T. Soderstron (1983) Theory and Practice of Recursive Identification, MIT Press.

Ljung, L. (1977), "Analysis of Recursive Stochastic Algorithms", IEEE Tran. Aut. Control, Vol. Ac-22, No. 4, pp. 551-575.

Reidle, B., and P. Kokotovic, "A Stability-Instability Boundary for Slow Adaptation and Unmodeled Dynamics", Proc. 23rd IEEE CDC, Las Vegas, Nevada, December 1984.

Shah, S.C., and M.A. Floyd, and L.L. Lehman (1985), "MATRIX$_X$: Control Design and Model Building CAE Capability," to appear, Advances in Computer-Aided Control Engineering, Eds. M. Jamshidi and C.J. Herget, North Holland, 1985.

Walker, R.A., S.C Shah, and N.K. Gupta (1984), "Computer-Aided-Engineering (CAE) for Systems Analysis, "IEEE Proceedings, Vol. No. 12, December, 1984.

APPENDIX A

An Input-Output View of Robustness in Adaptive Control

An Input Output View of Robustness in Adaptive Control*

R. L. Kosut† and C. R. Johnson, Jr‡

An input output theory of adaptive control provides a means of determining the robustness properties of adaptive algorithms.

Key Words—Adaptive control, robustness, robust control, stability, model reference adaptive control.

Abstract—The stability and robustness properties of adaptive control systems are examined using input output stability theory, i.e. passivity and small-gain theory. A generic adaptive error system is developed based on the concept of a tuned system—an ideal converged (nonadaptive) closed-loop system. Using this error system with passivity theory gives conditions for global stability where only boundedness (in norm) is required on the external inputs, e.g. disturbance, reference and initial conditions. Small gain theory is used to develop local stability results where the magnitudes of the external inputs are restricted. In the global case a particular system operator (not the plant) is required to be strictly-passive, a condition which is unlikely to hold in actual use due to unmodeled dynamics. The local results, however, are not so restricted and allow for unmodeled dynamics. In this latter case an estimate of the stability margin is given under a persistent excitation condition.

1. INTRODUCTION

AT A VERY basic level, the issues involved in adaptive control design are no different from nonadaptive (robust) control design. In either case the goal is to maintain specified performance properties despite uncertainty about the dynamics of the plant to be controlled, as well as uncertainty about its environment. In the nonadaptive case the problem of robustness to unmodeled dynamics is well formulated (e.g. Doyle and Stein, 1981; Zames and Francis, 1983). However, research in adaptive control theory has focused almost exclusively on the case where the plant can be fully represented by some member of a family of linear finite-dimensional parametric models (e.g. Narendra, Lin and Valavani, 1980; Goodwin, Ramadge and Caines, 1980). Thus, the model error due to unmodeled dynamics is presumed to be zero. Unfortunately, unmodeled dynamics can cause adaptive controllers to exhibit significant performance degradation and instability, even with an initial controller parameterization that closely approximates the desired closed-loop response (Rohrs and co-workers, 1981, 1982; Ioannou and Kokotovic, 1983a,b). These simulated circumstances of undesirable behavior are in sharp contrast with successful applications of adaptive control where reduced-order modeling is unavoidable (e.g. Astrom, 1983). This issue of model error, then, is of undeniable practical importance, because no actual plant is truly linear and finite-dimensional.

Perhaps the main reason for the lack of a robust adaptive control theory is that the emphasis has been on *global* results. What we mean by 'global' is that the intent is to require as little *a priori* information about the plant parametrization and the external inputs as possible to prove stable behavior. Because of this, the resulting requirements (i.e. assumptions) are too strong, e.g. known plant order. Therefore, it is compelling to abandon the requirement of global stability—a requirement that may well be beyond the needs of most actual systems—and develop conditions for *local* stability. The term 'local' is used in the sense that the plant uncertainty and external inputs are limited in a defined way, e.g. by restricting the magnitude and spectrum of the reference commands and disturbances, as well as the initial adaptive parameter error.

In this paper we will present an input output view§ of robustness in adaptive control. In particular, we shall draw attention to uncertain unmodeled plant dynamics—often referred to as model error—and to uncertain, but bounded, disturbances. Based on this view it may be possible to merge robust control theory with adaptive control theory.

The next section (Section 2) formalizes the conversion of a generic adaptive controller to an equivalent generic error system. The input output

* Received 6 November 1983; revised 4 April 1984. The original version of this paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by guest editor L. Ljung.

† Integrated Systems, Inc., 151 University Avenue, Palo Alto, CA 94301, U.S.A.

‡ School of Electrical Engineering, Cornell University, Ithaca, NY 14853, U.S.A.

§ A good source book on this material is the text by Desoer and Vidyasagar (1975). The notation used there is also used throughout this paper.

properties of the error system relate the performance of the nominal control system to that of the adaptive control system. Section 3 applies this formulation for a specific continuous-time adaptive model following algorithm. This permits the application in Section 4 of stability results for the continuous-time version of the generic error system. This section also includes a discussion of the strictly positive real (SPR) condition imposed on an operator within this error system. Finally, in Section 5, we will examine the issues involved in obtaining conditions for local stability and robustness. Though this paper concentrates on continuous-time systems (due to space limitations), this same input–output approach is applicable to robustness analysis of discrete-time adaptive control (e.g. Kosut, Johnson and Anderson, 1983; Ortega and Landau, 1983) as well as time-varying systems (Giomarr and Caines, 1984).

2. ERROR SYSTEM FORMULATION

(A) *Tuned control concept*

Consider the nonadaptive control system of Fig. 1, described by

$$e = H_{ew}(\pi)w \tag{1}$$

where $e(t) \in R^{n_e}$ is the error output, $w(t) \in R^{n_w}$ is the external input, and $\pi \in R^{n_\pi}$ is a set of controller parameters to be selected. For our purposes, $H_{ew}(\cdot)$ represents a closed-loop parametric feedback system dependent on the adjustable parameters in π. The output e of $H_{ew}(\cdot)$ is the error the control system experiences in meeting its objective given the external input w. Portions of $H_{ew}(\cdot)$ are not entirely known, e.g. the open-loop plant imbedded in $H_{ew}(\cdot)$. The input $w(t)$ is also not entirely known but can be assumed to be in a subset W of bounded signals. For example, $w(t)$ can consist of a set of reference commands and bounded disturbances. If the imbedded controller were adaptive, it would adjust π continuously on-line so as to reduce the error; but for now assume that π is constant and will be selected off-line.

If the control designer had all the 'off-line' time in the world to 'fiddle' with the parameters π, then it is hoped that a satisfactory adjustment would be obtained. Many strategies can be envisioned for determining a satisfactory π. In fact, such a satisfactory parameterization may not be unique

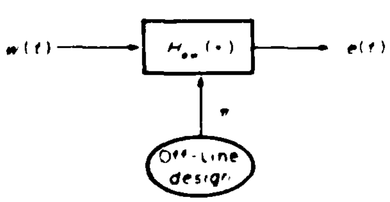


FIG. 1. Nonadaptive system.

but rather be any member of a set S, e.g.

$$S = \{\pi \in R^{n_\pi} : H_{ew}(\pi) \text{ has desired properties}\}. \tag{2}$$

Certain parameter sets S correspond to well-defined design strategies. Specifically

Matched. Let S denote the matched parameter set, i.e.

$$S = \{\pi \in R^{n_\pi} : H_{ew}(\pi) = 0\}. \tag{3}$$

Robust. Let S^0 denote the robust parameter set, i.e.

$$S^0 = \{\pi^0 \in R^{n_\pi} : \|H_{ew}(\pi^0)w\|_q \le p^0, \forall w(\cdot) \in W\} \tag{4a}$$

where the norm $\|\cdot\|_q$ is defined on the underlying function space. The finite constant p^0 represents the robust performance specification. Note that S^0 includes the 'optimal' robust solution, i.e. those $\pi \in R^{n_\pi}$ that solve

$$\inf_{\pi} \sup_{w} (\|H_{ew}(\pi)w\|_q / \|w\|_q). \tag{4b}$$

Tuned. Let $S^*_{w(\cdot)}$ denote the tuned parameter set associated with a particular $w(\cdot) \in W$, i.e.

$$S^*_{w(\cdot)} = \{\pi^*_{w(\cdot)} \in R^{n_\pi} : \|H_{ew}(\pi^*_{w(\cdot)})w\|_q \le p^*\}. \tag{5a}$$

The finite constant p^* represents a tuned performance specification. In order for (4) and (5a) to be meaningful, it is necessary that

$$p^* < p^0 \tag{5b}$$

i.e. the desired tuned performance is better than the desired robust performance. Also, $S^*_{w(\cdot)}$ will include the 'optimal' tuned solution, i.e. for each $w \in W$, those $\pi \in R^{n_\pi}$ that solve

$$\inf_{\pi} (\|H_{ew}(\pi)w\|_q / \|w\|_q). \tag{5c}$$

Ideally, the adaptive control should converge to the optimal parametrization of (5c). Thus, the tuned parameter set, denoted by S^*, is given by

$$S^* = \bigcup_{w(\cdot) \in W} S^*_{w(\cdot)}. \tag{6}$$

Note that each element of S^* is satisfactory for a particular $w(\cdot) \in W$ and that no one element in the subset $S^*_{w(\cdot)} \subset S^*$ need provide satisfactory control for a different $w(\cdot)$. (Although $\pi^*_{w(\cdot)} \in S^*$ emphatically denotes the dependence of the tuned parameters on $w(\cdot)$, we will henceforth denote membership in S^* by the simpler notation $\pi^* \in S^*$, where the $w(\cdot)$ dependence is to be implied.)

The error signal corresponding to the matched case is identically zero. It is this particular case that has received practically all the attention in adaptive control research and about which the strongest theoretical results are available. Unfortunately, in the first place this case excludes unmeasurable bounded disturbances which are a virtual certainty in any actual system. By unmeasurable bounded disturbances we mean those disturbances which can not be totally rejected at the output of the plant. In the second place, there will always be unmodeled dynamics, i.e. there are never enough parameters in π to solve $H_{ew}(\pi) = 0$ in practice. These remarks apply equally in a stochastic environment. For example whereas in the deterministic case $e = 0$, in the stochastic case $E\{e\} = 0$, with $E\{\cdot\}$ the expectation operator. Thus, the unmeasurable bounded disturbances alluded to above have their stochastic counterpart as processes which do *not* have zero mean, i.e. $E\{e\} \neq 0$ for any π.

The more appealing of the other two sets is the tuned set S^*, defined in (6). The associated error signal

$$e^* = H_{ew}(\pi^*)w \tag{7}$$

is referred to as the *tuned error* and $H(\pi^*)$ as the *tuned system*. Although $e^*(t) = 0$ is ruled out due to the impracticality of $\pi^* \in S$, we do not preclude the case where $e^*(t) \to 0$. This latter case still presumes a degree of idealization. Consider the case where the external input w consists of a step reference command with no disturbance and e^* is the difference between the plant output and the reference command. Thus, $e^*(t) \to 0$ is the ideal output error for any stabilizing controller engendering unit d.c. gain. This class of tuned controllers can be quite large even if $\dim(\pi^*) < \dim(\bar{\pi})$. Now consider the impact of a bounded disturbance, which is not necessarily of any particular functional form, such as a broadband bounded signal. Clearly, with such bounded disturbances present, $e^*(t) \nrightarrow 0$, and can only be assumed to be bounded.

An important comparison for the tuned set S^* is to the robust set S^0 (4). Let

$$e^0 = H_{ew}(\pi^0)w \tag{8}$$

denote the *robust error*. Recall from (5) and (6) that the tuned parameters π^* are dependent on a particular $w(\cdot) \in W$, whereas the robust parameters π^0 are not. Hence, the tuned error can never exceed the robust error, i.e. for a particular $w(\cdot) \in W$,

$$\|e^*\| = \|H_{ew}(\pi^*)w\| \leq \|e^0\| = \|H_{ew}(\pi^0)w\|. \tag{9}$$

Condition (9) also follows from the fact that $p^* < p^0$ (5b). Note that it is possible for the robust set S^0 to be empty even though the tuned set S^* is not. If S^0 is not empty, then consideration of an adaptive controller is justified if for some 'large' subset of $w(\cdot) \in W$, various tuned controllers exist such that *each* engenders

$$\|e^*\| < \|e^0\|. \tag{10}$$

If this were not the case, then a robust controller would suffice. This requirement (10) is weaker than the requirement $p^* < p^0$, which may not be attainable for all $w(\cdot) \in W$, since (10) is required only over a subset of W. However, even if (10) holds, adaptation may cause the error during adaptation to become either excessive or to otherwise exceed specifications.

The usefulness of defining the tuned parameter set will be borne out in the next subsection. The tuned set is used there to develop a generic adaptive error system. At this point, however, we remark that it is not necessary to solve the optimization problem defined implicitly in (5), rather we only need to know that a solution exists which is better than the robust solution (4).

(B) *Adaptive error system*

Now consider the adaptive version of (1), depicted in Fig. 2, and described by the input–output relations

$$\begin{bmatrix} e \\ \xi \end{bmatrix} = \begin{bmatrix} H_{ew}(\hat{\pi}) \\ H_{\xi w}(\hat{\pi}) \end{bmatrix} w = H(\hat{\pi})w \tag{11a}$$

$$\hat{\pi} = \Omega[\hat{\pi}(0), e, \xi] \tag{11b}$$

where $\hat{\pi}(t) \in R^{n_\pi}$ are the *adaptation parameters* which are generated from the parameter *adaptive algorithm* Ω, and $\hat{\pi}(0) \in R^{n_\pi}$ is the initial parameter estimate. The adaptive algorithm is driven by the output or *adaptation error* $e(t) \in R^{n_e}$ and the *regressor* $\xi(t) \in R^{n_\xi}$. The regressor is obtained from sensed signals within the feedback system.

We want to ultimately determine conditions under which the adaptive system (11) is stably attracted to the set of tuned systems (6). Recall that the tuned system set is likely to contain more than a single member, thus by stability we mean stability 'about' a (possibly disconnected) set rather than about a point.

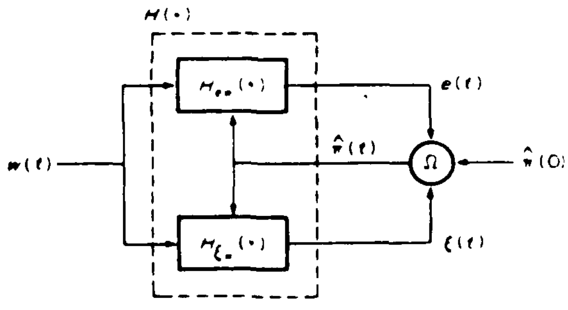


FIG. 2. Adaptive system.

The analysis is facilitated by transforming the format of Fig. 2 into an *error system* format. To do this we must define the structure of the *adaptive control*. Consider a single-input single-output (SISO) plant embedded in $H(\cdot)$ whose input $u(t)$ is given by the *bilinear* expression

$$u(t) = -\xi(t)'\hat{\pi}(t) \tag{12}$$

Note that this development is not limited to SISO plants. The extension of (12) to the multivariable case involves a similar expression for each control channel, i.e.

$$u_i(t) = -\xi_i(t)'\hat{\pi}_i(t), \quad i = 1, \dots, n_u \tag{13}$$

where $\xi_i(t)$ and $\hat{\pi}_i(t)$ are vectors consisting of elements from the regressor and parameter vectors, respectively. However, only the SISO case will be considered here to reduce the complexity of the development and allow sharper focus on the adaptive systems issues. Normally, $\xi(t)$ consists of the plant inputs and outputs, or filtered versions thereof. For example, in discrete-time systems $\xi(t)$ consists of a finite record of past plant inputs and outputs.

Although the bilinear structure in (12) and (13) remains the most widely used and studied format, nonetheless, other structures (as yet underdeveloped) may be more suitable to certain problems e.g. distributed and or nonlinear structures.

We will now make a strong assumption regarding the way in which $u(t)$ and $w(t)$ are transmitted through $H(\cdot)$ into $e(t)$ and $\xi(t)$.

Assumption. The map $(w, u) \mapsto (e, \xi)$ is linear time-invariant (LTI), i.e.

$$\begin{bmatrix} e(t) \\ \xi(t) \end{bmatrix} = \begin{bmatrix} G_{ew}(s) & G_{eu}(s) \\ G_{\xi w}(s) & G_{\xi u}(s) \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} = G(s) \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \tag{14}$$

where $G(s)$ is the *open-loop interconnection* matrix whose elements are proper rational functions. (To simplify notation we will use s to denote either the Laplace transform variable or the differential operator, depending on the context.)

The adaptive system (11) with bilinear control (13) and LTI interconnections (14) is shown in Fig. 3. To transform this system to an error system, define the *parameter error*

$$\tilde{\pi}(t) = \hat{\pi}(t) - \pi^* \tag{15a}$$

with

$$\pi^* \in S^* \tag{15b}$$

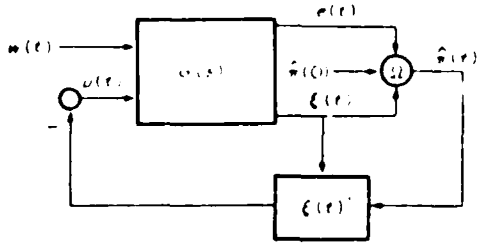


Fig. 3. Adaptive system with LTI interconnection.

and the *adaptive control error*

$$r(t) = \xi(t)'\tilde{\pi}(t). \tag{16}$$

Thus, Fig. 3 can be redrawn as shown in Fig. 4 and described by

$$\begin{bmatrix} e(t) \\ \xi(t) \end{bmatrix} = -G(s)\begin{bmatrix} 0 \\ \xi(t)'\pi^* \end{bmatrix} + G(s)\begin{bmatrix} w(t) \\ -r(t) \end{bmatrix}. \tag{17a}$$

Hence,

$$\begin{bmatrix} e(t) \\ \xi(t) \end{bmatrix} = \begin{bmatrix} H^*_{ew}(s) & H^*_{er}(s) \\ H^*_{\xi w}(s) & H^*_{\xi r}(s) \end{bmatrix}\begin{bmatrix} w(t) \\ -r(t) \end{bmatrix} = H^*(s)\begin{bmatrix} w(t) \\ -r(t) \end{bmatrix} \tag{18a}$$

where

$$H^*_{ew}(s) = G_{ew}(s) + G_{eu}(s)\pi^{*\prime}(I + G_{\xi u}(s)\pi^{*\prime})^{-1}G_{\xi w}(s) \tag{18b}$$

$$H^*_{er}(s) = G_{eu}(s) + G_{eu}(s)\pi^{*\prime}(I + G_{\xi u}(s)\pi^{*\prime})^{-1}G_{\xi u}(s) \tag{18c}$$

$$H^*_{\xi w}(s) = (I + G_{\xi u}(s)\pi^{*\prime})^{-1}G_{\xi w}(s) \tag{18d}$$

$$H^*_{\xi r}(s) = (I + G_{\xi u}(s)\pi^{*\prime})^{-1}G_{\xi u}(s). \tag{18e}$$

The dashed box in Fig. 4 is $H^*(s)$. We will refer to $H(s)$ as the *tuned interconnections*. Note that the *tuned error* (7) is identical to

$$e^*(t) = H^*_{ew}(s)\,w(t). \tag{19}$$

We also make use of the *tuned regressor*, defined as

$$\xi^*(t) = H^*_{\xi w}(s)\,w(t). \tag{20}$$

Finally, the error system (Fig. 4) can be depicted as in Fig. 5, where

$$e(t) = e^*(t) - H^*_{er}(s)\,r(t) \tag{21a}$$

$$\xi(t) = \xi^*(t) - H^*_{\xi r}(s)\,r(t) \tag{21b}$$

$$r(t) = \xi(t)'\tilde{\pi}(t) \tag{21c}$$

$$\hat{\pi}(t) = \Omega[\hat{\pi}(0), e(\cdot), \xi(\cdot)] \tag{21d}$$

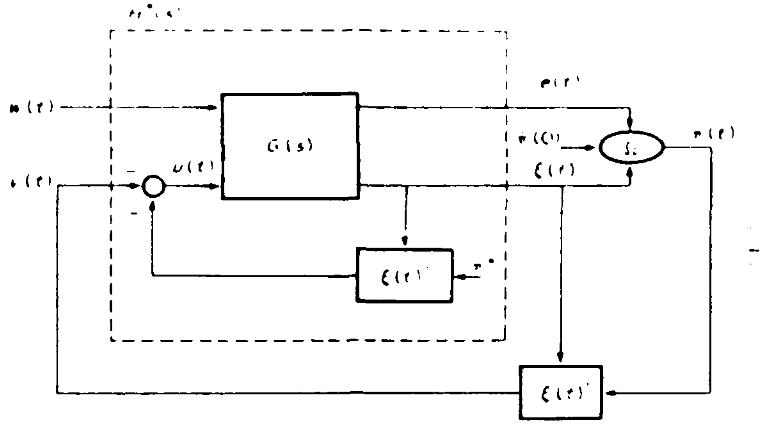


Fig. 4. Adaptive error system.

Figure 5 reveals that this error system is composed of a nonlinear system in the forward loop, denoted by N_{ve}, and the LTI system $H^*_{e}(s)$ in the feedback path. Thus, the error system is driven externally by the tuned system outputs $e^*(t)$ and $\zeta^*(t)$, and the initial parameter error $\tilde{\pi}(0) = \hat{\pi}(0) - \pi^*$.

(C) Existence of the tuned controller

The designation of the tuned controller is the concept most important to extracting a meaningful error system from the description of an adaptively controlled system. It might appear that the ability to specify this tuned controller presupposes our knowledge of an acceptable solution to the underlying adaptive control problem. This is not entirely the case. Given the parametric controller structure of $C(\cdot)$, we need only have an *approximate a priori* knowledge of the system behavior. Given a particular π^*, we will discover that the restrictions on H^*_{e} and H^*_{ζ} can be assessed from knowledge of the tuned controller and bounds on the magnitude of the plant modeling error. Such information is a practical result of a thorough plant modeling study. Thus, the study of the stability of (21) will have practical meaningfulness. We will examine a particular continuous-time adaptive controller in the following section and derive the form of H^*_{e} and H^*_{ζ}.

3. CONTINUOUS-TIME ADAPTIVE MODEL-FOLLOWING

To characterize H^*_{e} and H^*_{ζ} some designation of a tuned controller must be provided. We make the choice by assuming that no modeling error exists in the nominal plant parametric model. We close with consideration of the degree of plant mismodeling allowed such that this tuned controller is robust, i.e. maintains stable control of the actual system. Following this discussion, in Section 4 we consider the effect of the adaptive algorithm.

(A) Direct model reference adaptive control

Consider the model reference adaptive control (MRAC) system shown in Fig. 6, described by

$$y(t) = d(t) + P(s)u(t) \quad \text{(plant)} \tag{22a}$$

$$\bar{y}(t) = \bar{H}(s)r(t) \quad \text{(reference model)} \tag{22b}$$

$$e(t) = y(t) - \bar{y}(t) \quad \text{(tracking error)} \tag{22c}$$

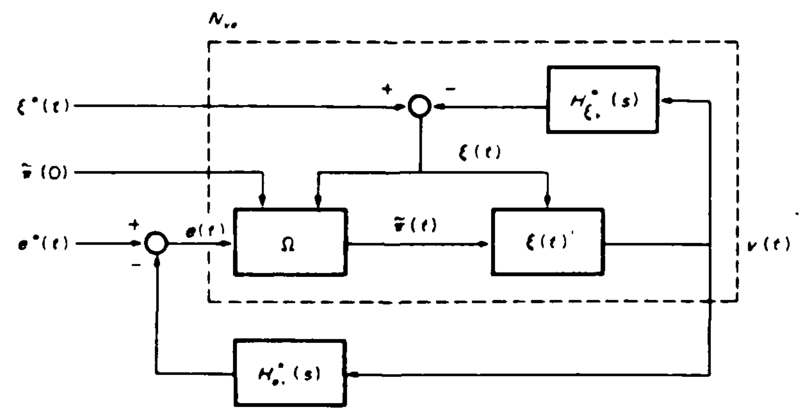


Fig. 5. Adaptive error system.

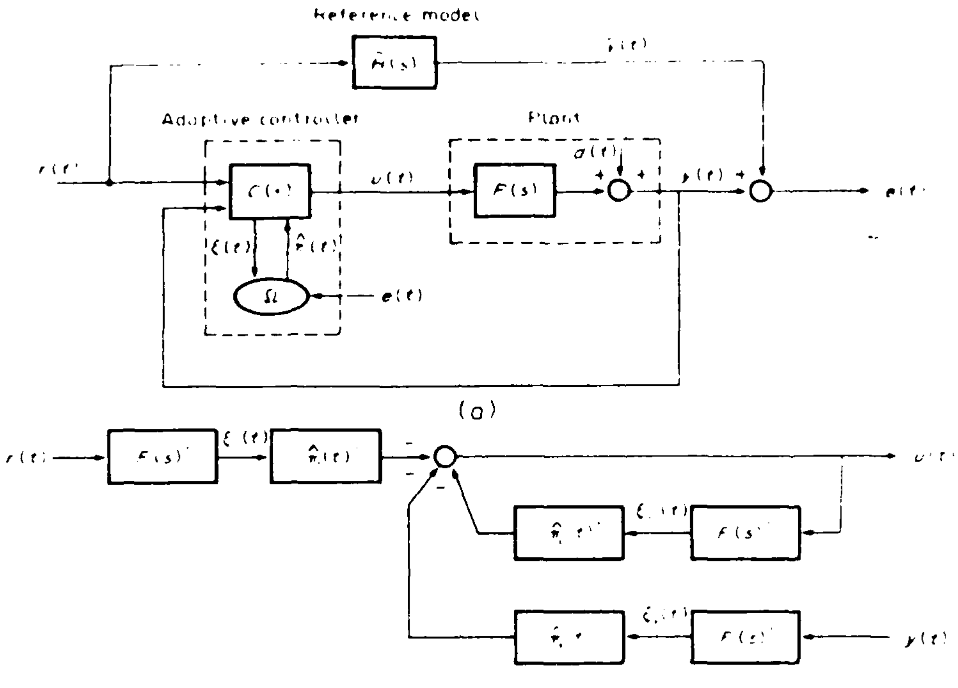


Fig. 6. Model reference adaptive system: (a) block diagram; (b) controller detail.

where $d(t)$ consists of disturbances and plant initial conditions, and $r(t)$ is the reference command. Let $\{C(\cdot),\Omega\}$ denote the adaptive controller, where Ω is the parameter adaptive algorithm and $C(\cdot)$ is the parametric controller. Following Narendra, Lin, and Valavani (1980), let $C(\cdot)$ have the bilinear form

$$\begin{aligned} u(t) &= -\xi(t)'\hat{\pi}(t) \\ &= -\xi_u(t)'\hat{\pi}_u(t) - \xi_y(t)'\hat{\pi}_y(t) - \xi_r(t)'\hat{\pi}_r(t) \quad (23a) \end{aligned}$$

where the regressor is given by filtered versions of u, y and r

$$\begin{aligned} \xi(t)' &= [\xi_u(t)', \xi_y(t)', \xi_r(t)']' \\ &= [F(s)u(t), F(s)y(t), -F(s)r(t)]' \quad (23b) \end{aligned}$$

with

$$F(s) = \left(\frac{1}{L(s)}, \ldots, \frac{s^{k-1}}{L(s)}\right) \quad (23c)$$

and

$$L(s) = s^k + \alpha_1 s^{k-1} + \ldots + \alpha_k. \quad (23d)$$

Thus, there are $3k$ adaptive parameters. Using the definition of adaptive control error in (21c), the MRAC control signal (22) can be expressed as

$$u(t) = -\frac{A_1^*(s)}{L(s)}u(t) - \frac{A_2^*(s)}{L(s)}y(t) + \frac{A_3^*(s)}{L(s)}r(t) - v(t) \quad (24)$$

where the tuned parametrization $\pi^*(=\hat{\pi}-\tilde{\pi})$ is distributed among the control elements as follows:

$$A_1^*(s) = \pi_1^* s^{k-1} + \ldots + \pi_k^* \quad (25a)$$

$$A_2^*(s) = \pi_{k+1}^* s^{k-1} + \ldots + \pi_{2k}^* \quad (25b)$$

$$A_3^*(s) = \pi_{2k+1}^* s^{k-1} + \ldots + \pi_{3k}^*. \quad (25c)$$

Thus, (24) becomes

$$u(t) = C_{ur}^*(s)\, r(t) - C_{uy}^*(s)\, y(t) - C_{uv}^*(s)v(t) \quad (26a)$$

where

$$C_{uy}^*(s) = \frac{A_2^*(s)}{L(s) + A_1^*(s)},$$

$$C_{ur}^*(s) = \frac{A_3^*(s)}{L(s) + A_1^*(s)},$$

$$C_{uv}^*(s) = \frac{L(s)}{L(s) + A_1^*(s)}. \quad (26b)$$

We will refer to $C^* = [C_{ur}^*, C_{uy}^*, C_{uv}^*]$ as the *tuned* controller. The adaptive error system (21) corresponding to the MRAC of Fig. 6 is shown in Fig. 5. The tuned signals (3.19) and (3.20) are

$$e^* = [(1 + PC_{uy}^*)^{-1}PC_{ur}^* - \bar{H}]\, r - (1 + PC_{uy}^*)^{-1}d \quad (27a)$$

$$\xi^* = \begin{bmatrix} (1 + PC_{uy}^*)^{-1}C_{ur}^*F'r - C_{uy}^*(1 + PC_{uy}^*)^{-1}F'd \\ (1 + PC_{uy}^*)^{-1}PC_{ur}^*F'r - (1 + PC_{uy}^*)^{-1}F'd \\ -F'r \end{bmatrix} \quad (27b)$$

The tuned interconnections (18) are

$$H^*_{ry} = (I + PC^*_{uy})^{-1}PC^*_{ur} \tag{28a}$$

$$H^* = \begin{bmatrix} (I + PC^*_{uy})^{-1}C^*_{ur}I \\ (I + PC^*_{uy})^{-1}PC^*_{ur}I \\ 0 \end{bmatrix} \tag{28b}$$

(B) Tuned system control design

There are any number of ways to design the tuned controller C^*. The important point — no matter what design method is used — is that the tuned design must be robust, because the plant $P(s)$ in (27) and (28) is not entirely known. Recall from (5) that the tuned controller is dependent on the plant. For example, the $3k$ parameters in π^* cannot make $r^* \rightarrow e$ in (27a) be identically zero. This can be viewed as a reduced order design problem or, as in the discussion that follows, a problem in robustness to unmodeled dynamics.

Suppose that the actual plant can be described by

$$P(s) = [1 + \Delta(s)]P^*(s) \tag{29a}$$

$$P^*(s) = \frac{b_0 B^*(s)}{A^*(s)}$$

$$= \frac{b_0(s^m + b_1 s^{m-1} + \ldots + b_m)}{s^n + a_1 s^{n-1} + \ldots + a_n}, \; m < n \tag{29b}$$

where $P^*(s)$ is a *tuned parametric model* of $P(s)$, i.e. the parameters $(b_0, \ldots, b_m, a_1, \ldots, a_n)$ provide a good fit, say at low frequencies. The transfer function $\Delta(s)$ represents unmodeled dynamics, i.e. those dynamics in $P(s)$ not accounted for by $P^*(s)$, e.g. high frequency efforts. Assume that $\Delta(s)$ is stable but is otherwise unknown except for a bound, i.e.

$$|\Delta(j\omega)| \leq \delta(\omega), \; \forall \omega \in R. \tag{29c}$$

This type of modeling uncertainty is said to be *unstructured* (Doyle and Stein, 1981). In more general terms, (29) provides a *set description* of the plant rather than a single parametric model, such as P^* (Safonov, 1980).

We will now examine the impact of model error on a tuned control design based only on the parametric model. The model reference format suggests that we make e^* as small as possible. To eliminate the tracking error term in (29a) entirely, we will use the procedure described in Egardt (1979), which requires that the following information is known:

(1) $n > m$ ($P^*(s)$ is strictly proper)
(2) n and m are known
(3) $B^*(s)$ has all zeros strictly inside the left half plane

Also, the reference model transfer function is assumed to be

$$\bar{H}(s) = \frac{\bar{B}(s)}{\bar{A}(s)} = \frac{\bar{b}_0 s^m + \bar{b}_1 s^{m-1} + \ldots + \bar{b}_m}{s^n + \bar{a}_1 s^{n-1} + \ldots + \bar{a}_n} \tag{30}$$

where $\bar{b}_0, \ldots, \bar{b}_m, \bar{a}_1, \ldots, \bar{a}_n$ are preselected constants, and where $\bar{H}(s)$ is exponentially-stable, i.e. all zeros of $\bar{A}(s)$ are strictly inside the left half plane. It is well to point out here that although assumption (2) above can be satisfied by the parametric model (29b), this is not the case for the actual plant (29a) due to the presence of the unstructured uncertainty (29c).

The tuned controller structure proposed in Egardt (1979) requires that

$$C^*_{uy} = \frac{R^*}{b_0 B^* S^*} \tag{31a}$$

$$C^*_{ur} = \frac{\bar{B}T^*}{b_0 B^* S^*} \tag{31b}$$

where T^* is a stable monic polynomial of degree $n_T > n - m - 1$, and where the polynomials S^* and R^* uniquely solve the polynomial equation

$$T^*\bar{A} = R^* + AS^* \tag{31c}$$

with S^* monic of degree n_T, and R^* of degree $n - 1$. With no model error ($\Delta = 0$), this controller (31), in addition to stabilizing the tuned system, also makes the transfer function from r into y^* identical to the reference model $\bar{H}(s)$. Thus, the tuning of (31) is for the subset of $\mathcal{W}$ composed of bounded reference signals and zero disturbance. The effect of (31) on C^*_{ur} will shortly be made apparent. Comparing (33) with (26) motivates solving for π^* from

$$L + A_1^* = B^*S^*$$
$$A_2^* = \frac{1}{b_0}R^* \tag{32}$$
$$A_3^* = \frac{1}{b_0}\bar{B}T^*.$$

A solution for π^* exists provided that

$$k = n_1 + m \geq n. \tag{33}$$

With this choice for (A_1^*, A_2^*, A_3^*), the tuned controller is given by (31) and by

$$C^*_{ur} = \frac{L}{BS^*}. \tag{34}$$

(C) The effect of model error on tuned system performance

It is convenient to define the transfer function

$$G^* = \frac{R^*}{T^*\bar{A}} \tag{35}$$

Using the tuned controller just described, the tuned signals (27) are then

$$e^* = (1+\Delta G^*)^{-1}\left[\Delta(1-G^*)Hr + \frac{A^*S^*}{\bar{A}T^*}d\right] \quad (36)$$

$$\xi^* = \begin{bmatrix} (1+\Delta G^*)^{-1}\left[\dfrac{A^*B}{b_0B^*\bar{A}}Lr + \dfrac{A^*R^*}{b_0B^*T^*\bar{A}}Ld\right] \\ (1+\Delta G^*)^{-1}\left[(1+\Delta)\dfrac{B}{\bar{A}}Lr - \dfrac{S^*A^*}{T^*\bar{A}}Ld\right] \\ -Lr \end{bmatrix} \quad (36b)$$

and the tuned interconnections (28) become

$$H_{e}^* = (1+\Delta G^*)^{-1}(1+\Delta)\frac{b_0L}{T^*\bar{A}} \quad (37a)$$

$$H_{\xi}^* = \begin{bmatrix} (1+\Delta G^*)^{-1}\dfrac{A^*L}{B^*T^*\bar{A}}I \\ (1+\Delta G^*)^{-1}(1+\Delta)\dfrac{b_0L}{T^*\bar{A}}F \\ 0 \end{bmatrix} \quad (37b)$$

The tuned system with no model error ($\Delta = 0$) is exponentially stable, since, by assumption, the poles of $(B^*)^{-1}$, $(\bar{A})^{-1}$, and $(T^*)^{-1}$ are in the open left half plane. Hence e^* and ξ^* are bounded if r and d are bounded. Thus, the stability of the *actual* tuned system is guaranteed if and only if

$$(1+\Delta G^*)^{-1} \text{ and } (1+\Delta G^*)^{-1}\Delta \text{ are exponentially stable.} \quad (38)$$

Note that under these conditions, the tuned interconnections in (37b) remain exponentially stable. However, it is not necessary (nor possible by assumption) to have a complete description of Δ in order to satisfy (38). For example, if Δ is known to be exponentially stable, then with G^* known to be exponentially stable, (38) holds if (e.g. Doyle and Stein, 1981)

$$|\Delta(j\omega)||G^*(j\omega)| < 1, \ \forall \omega \in R. \quad (39)$$

Satisfaction of (39) requires that

$$|\Delta(j\omega)| < \delta(\omega) = 1/|G^*(j\omega)|, \ \forall \omega \in R. \quad (40)$$

We will show in Section 4 that $\delta(\omega) < 1$ is the limit imposed on $\delta(\omega)$ by the usual global stability results for continuous-time adaptive systems. Similar limits are also encountered with discrete-time adaptive systems.

4. GLOBAL STABILITY CONDITIONS

The purpose of this section is to introduce global stability conditions applicable to the generic error system of (21). In the preceding section we specified an adaptive controller structure $C(\cdot)$ from which we then developed the tuned system $(r, d) \to (e^*, \xi^*)$ and the interconnection operators H_e^* and H_ξ^*. We now need to characterize the adaptive law Ω in (21d). With this connection we will be able to interpret some conditions under which such a continuous-time adaptive controller possesses a (limited) degree of robustness. Our interpretive remarks will address the restrictiveness of the SPR condition on H_e^* that arises in practically all global stability theorems.

(A) *The adaptive algorithm*

We will begin by specifying the adaptive law(s) of interest. A large class of adaptive algorithms (21d) have the from

$$\dot{\hat{\pi}}(t) = A[\xi(\cdot), \omega(t)], \ \hat{\pi}(0) \in R^p \quad (41a)$$

$$\omega(t) = \xi(t)\, e(t). \quad (41b)$$

We will refer to $A(\ldots)$ as the *adaptation gain*, which is a nonlinear operator. In general $A[\cdot,\cdot]$ can have memory, usually only in $\xi(t)$. The adaptive algorithm can also be expressed in terms of the parameter error $\tilde{\pi}(t)$ as

$$\dot{\tilde{\pi}}(t) = A[\xi(\cdot), \omega(t)], \ \tilde{\pi}(0) = \hat{\pi}(0) - \pi^*. \quad (42)$$

The complete adaptive error system (21), including the adaptive algorithm (42), is shown in Fig. 7.

The choice of algorithms, i.e. the variety of proposed adaptation gains, is virtually unlimited. The following two are our chosen representatives:

Constant gain (Narendra, Lin, and Valavani, 1980).

$$A[\xi(\cdot), \omega(t)] = A_0\omega(t) \quad \text{where } A_0 \in R^{p\times p}, A_0 = A_0' > 0. \quad (43)$$

Retarded gain (Kreisselmeier and Narendra, 1982)

$$A[\xi(\cdot), \omega(t)] = \begin{cases} A_0\omega(t), & |\tilde{\pi}(t)| < c \\ A_0[\omega(t) - (1 - |\tilde{\pi}(t)|/c)^2\tilde{\pi}(t)], & |\tilde{\pi}(t)| \geq c \end{cases} \quad (44)$$

where $A_0 \in R^{p\times p}$, $A_0 = A_0' > 0$, and $c \geq \max|\pi^*|$.

We will use the concept of persistent excitation that has proven important in adaptive control, as well as in adaptive system identification.

Definition (Anderson, 1977). A function $f(\cdot)$: $R_+ \to R^n$ is *persistently exciting*, denoted by

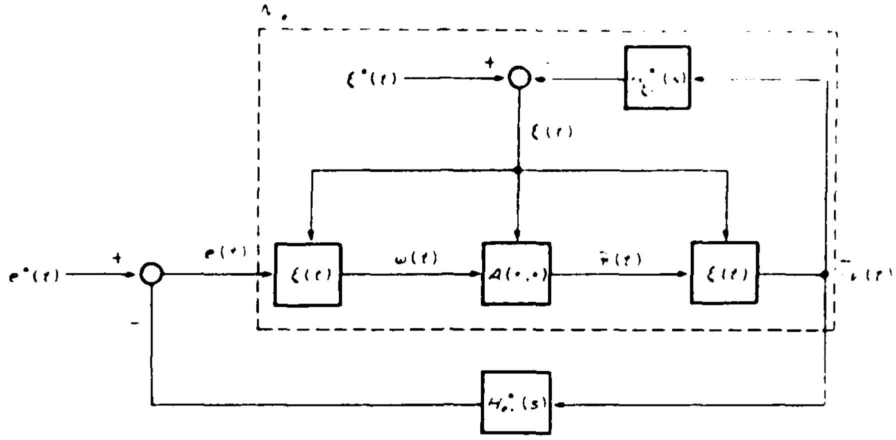


Fig. 7. MRAC error system.

$f \in$ PE, if there exists positive constants α_1, α_2 and α_3 such that

$$\alpha_2 I_n \geq \int_s^{s+\alpha_3} f(t) f(t)' dt \geq \alpha_1 I_n, \forall s \in R_+. \quad (45)$$

We will discuss the implications of persistent excitation on global stability below, as well as in Section 5, in regard to local stability.

(B) A global stability theorem

Theorem 1, which follows, gives conditions for global stability of the adaptive error system of Fig. 7. The term 'global' refers to the intention of seeking the minimal (reasonable) restrictions on the tuned signals $e^*(t)$ and $\xi^*(t)$, and the tuned interconnections $H^*_{e\xi}(s)$ and $H^*_{\xi\xi}(s)$ resulting in the proof that e and ξ remain bounded, i.e. (21) is stable, for any finite $\tilde{\pi}(0)$. (A detailed proof of Theorem 1 is given in Kosut and Friedlander (1983).) In particular, we will consider the following two tuned system signal sets as 'inputs' to the error system:

$$W^*_0 = \{e^*, \xi^*, \tilde{\pi}(0) | e^*, \dot{e}^* \in L_2 \cap L_\infty, \xi^* \in L^r_\infty, \tilde{\pi}(0) \in R^r\} \quad (46)$$

$$W^*_B = \{e^*, \xi^*, \tilde{\pi}(0) | e^* \in L_\infty, \xi^* \in L^r_\infty, \tilde{\pi}(0) \in R^r\}. \quad (47)$$

Note that e^*, $\dot{e}^* \in L_2 \cap L_\infty$ essentially implies boundedness and ultimate decay to zero, whereas inclusion in L_∞ only implies boundedness.

Theorem 1. For the system of Fig. 7, assume that

(A1) $H^*_{\xi\xi}(s)$ is strictly proper and exponentially stable (48)

(A2) $H^*_{e\xi}(s)$ is strictly positive real (SPR), i.e. $H^*_{e\xi}(s)$ is strictly proper, exponentially stable, and there exists a finite constant $\rho > 0$ such that

$$\mathrm{Re}\,[H^*_{e\xi}(j\omega)] \geq \rho |H^*_{e\xi}(j\omega)|^2, \forall \omega \in [0, \infty). \quad (49)$$

Under these conditions, algorithms (43) or (44) result in the following properties:

(i) If $(e^*, \xi^*, \tilde{\pi}(0)) \in W^*_0$ then $\tilde{\pi}$, e, ξ, and r are bounded (in L_∞), $\dot{\tilde{\pi}}(t) \to 0$, and $e^*(t) - e(t) \to 0$. In addition, if $\xi^* \in$ PE, then $\tilde{\pi}(t) \to 0$ exponentially.

(ii) If $(e^*, \xi^*, \tilde{\pi}(0)) \in W^*_B$ and $\xi \in$ PE, then $\tilde{\pi}$, e, ξ, and r are bounded (in L_∞).

(iii) If $(e^*, \xi^*, \tilde{\pi}(0)) \in W^*_B$ and $\xi \in$ PE, then the results of (ii) still follow by using the gain algorithm (44).

Comments on Theorem 1. Though theoretically significant, these results do not offer the engineering design guidelines we would like to obtain. The major reason is that $H^*_{e\xi}(s) \in$ SPR (condition (49)) is virtually impossible to achieve for any actual system. The primary culprit here is the effect of unmodeled dynamics. Details on this issue may be found in Rohrs and co-workers (1982). Further discussion will be provided in the following subsection.

Another technical hurdle is that the only realistic case, insofar as the tuned signals (e^*, ξ^*) are concerned, is when $(e^*, \xi^*) \in W^*_B$. This is the situation induced by continual bounded disturbances, such as would normally be encountered. But in this case the theory requires that either $\xi \in$ PE as in part (ii) or that the adaptation gain is retarded as in part (iii). With bounded disturbances present it is not known how to guarantee $\xi \in$ PE, since ξ is generated *inside* the adaptive loop. Note that part (i) only requires that the tuned regressor $\xi^* \in$ PE rather than the actual regressor $\xi \in$ PE as in part (ii). However, this requires $(e^*, \xi^*) \in W^*_0$, which is only possible when the control structure provides asymptotic model following and disturbance rejection. This is the classic case studied in the literature. Obviously, unmodeled dynamics and bounded disturbances eliminate this ideal situation. A further difficulty regarding $\zeta \in$ PE is that this occurs at the expense of any set-point regulation, which deteriorates in the presence of PE signals. Using gain retardation does not require persistent excitation, but does require some *a priori* information, i.e. as in (44), the foreknowledge of an upper bound on $|\pi^*|$, which is not too difficult to obtain. Although retardation does handle bounded disturbance, the SPR condition is still required.

(C) *In pursuit of the SPR condition*

> '*When a man points to the stars, only a fool looks at his finger.*'
>
> *Anonymous.*

The intent of this aphorism is to divert any lingering anxieties about the SPR condition. It – the SPR condition – simply will not do as a major building block in adaptive control theory. But that does not mean a total abandonment of our aim: it suggests, rather, a redirection. We should be establishing a different path to the 'stars.' For now, however, we will remain earthbound and address the restrictiveness of the SPR requirement.

A necessary condition for $H_{e\zeta}^* \in$ SPR is that $H_{e\zeta}^*(s)$ have a relative degree of one. As pointed out by Rohrs and co-workers (1982), this imposes the requirement that the relative degree of the plant is known, e.g. examine the effect on the plant P in (28a). This knowledge, however, is unavailable due to the presence of unmodeled dynamics, as assumed in (29).

The same type of restriction can also be seen as follows. From (37a)

$$H_{e\zeta}^* = (1 + \Delta G^*)^{-1}(1 + \Delta)\bar{H}_{e\zeta}^* \qquad (50a)$$

$$\bar{H}_{e\zeta}^* = \frac{b_0 L}{T^* \bar{A}} \qquad (50b)$$

If Δ is exponentially stable, but is otherwise unstructured, then conditions for $H_{e\zeta}^* \in$ SPR include (Kosut and Friedlander, 1983)

$$(1)\ \bar{H}_{e\zeta}^* \in \text{SPR} \qquad (51a)$$

$$(2)\ |\Delta(j\omega)| < 1. \qquad (51b)$$

Since $\bar{H}_{e\zeta}^*$ is dependent only on the parametric model P^*, it is not difficult to find π^* such that $\bar{H}_{e\zeta}^* \in$ SPR. Unfortunately, the drawback is that (51b) is a condition that is almost surely violated, due to typically unmodeled high frequency dynamics.

So if $H_{e\zeta}^* \in$ SPR can never hold, can we eliminate the SPR requirement or add some clever filtering to desirably alter $H_{e\zeta}^*$? For the perfect modeling case ($\Delta = 0$) it is possible to obtain (Monopoli, 1974; Landau, 1978; Egardt, 1979)

$$H_{e\zeta(\Delta=0)}^* = H_{e\zeta}^* = \text{positive constant} \qquad (52)$$

Although a positive constant is SPR (and hence, satisfies (51)), condition (51b) is still required for $H_{e\zeta}^*(\Delta \neq 0)$ to be SPR.

These disclaimers lead us away from the global approach typified by Theorem 1 to the establishment of local stability results which are robust to unmodeled dynamics and bounded disturbances.

5. LOCAL STABILITY CONDITIONS

In this section we indicate a means of obtaining local stability conditions. To clarify the distinction between local and global, consider, for example, result (iii) of Theorem 1. This result holds if $H_{e\zeta}^* \in$ SPR, H_{ζ}^* exponentially stable, $(e^*, \zeta^*) \in \mathcal{B}_{\infty}^n$, $\zeta \in$ PE, and $|\tilde{\pi}(0)| < \infty$. Aside from the difficulties in establishing SPR and PE, all the conditions are virtually free of any magnitude constraints, and hence, are 'global' conditions. In every practical case, it is more than likely that magnitude information is available, e.g. *a priori* bounds on $\|e^*\|_\infty$, $\|\zeta^*\|_\infty$ and $|\tilde{\pi}(0)|$, as well as a bound on the gains of $H_{e\zeta}^*$ and H_{ζ}^*. For example, Egardt (1979) shows robustness properties for minimum phase systems with bounded output disturbances. Dead-zone and projection mechanisms can handle small unmodeled dynamics as shown by Praly (1983) and Samson (1983). Ioannou and Kokotovic (1983a,b) are able to give an estimate of the region of attraction without SPR or PE in the case of high frequency parasitics. Persistent excitation, and the resulting exponential stability property (see equation (62) in this section) also leads to robustness (e.g. Anderson and Johnson, 1982a,b; Anderson and Johnstone, 1983; Kosut, 1983). Various other gain normalizations have also been suggested (e.g. Gawthrop and Lim, 1982; Ortega and Landau, 1983). These theoretical results remain incomplete, because they do not as yet provide a useful means of assessing the impact of unmodeled dynamics, e.g. a frequency domain bound on model error, dependent on the 'return-difference gain' (e.g. Doyle and Stein, 1981).

In this section we will show in Theorem 2, under mild magnitude bounds, that the adaptive system is (locally) L_∞-stable. This result is quite general because the conditions are independent of the nature of the adaptive algorithm, e.g. dead-zones, normalizations, or persistent excitation.

To facilitate the analysis we will only consider the continuous-time error system (21) with constant gain adaptation alorithm (43). It is convenient to transform (21) to the following variational form, which is more useful for local analysis

$$\tilde{x} = \tilde{x}_l + \tilde{x}_{Nl} \tag{53a}$$

$$\tilde{x}_{Nl} = F f(\tilde{x}) \tag{53b}$$

where

$$\tilde{x} = (\tilde{\pi}, \tilde{e}, \tilde{\xi}) = (\pi - \pi^*, e - e^*, \xi - \xi^*) \tag{53c}$$

$$\tilde{x}_l = (\tilde{\pi}_l, \tilde{e}_l, \tilde{\xi}_l), f(\tilde{x}) = (\tilde{\xi}\tilde{\pi}, \tilde{\xi}\tilde{e}). \tag{53d}$$

Details on transforming (21) to (53) are in Kosut (1983). This form of the adaptive error system is obtained by linearization of (21) about e^*, ξ^* and π^*. The linearized perturbation response is $\tilde{x}_l$, almost identical to the linearized system studied by Rohrs and co-workers (1981), which was arrived at by a 'final approach analysis.' The remaining nonlinear terms $\tilde{x}_{Nl}$ are contained in $f(\tilde{x})$, a memoryless nonlinearity, and in F, a time-varying linear operator. The characteristics of F, as well as those of $\tilde{x}_l$, depend on the adaptation gain and the behavior of the tuned signals, e^* and ξ^*. For example, with the constant gain algorithm (43), the linearized perturbation response is

$$\tilde{\pi}_L = (I + LM)^{-1}\tilde{\pi}_0 + K\xi^* e^* \tag{54a}$$

$$\tilde{e}_L = -H^*_{e_i}\xi^{*\prime}\tilde{\pi}_L \tag{54b}$$

$$\tilde{\xi}_L = -H^*_{\xi_i}\xi^{*\prime}\tilde{\pi}_L \tag{54c}$$

with

$$F = \begin{bmatrix} KN & -K \\ H^*_{e_i}(1 - \xi^{*\prime}KN) & H^*_{e_i}\xi^{*\prime}K \\ H^*_{\xi_i}(1 - \xi^{*\prime}KN) & H^*_{\xi_i}\xi^{*\prime}K \end{bmatrix} \tag{55}$$

and where

$$L = \frac{1}{s}A_0 \tag{56a}$$

$$K = (I + LM)^{-1}L \tag{56}$$

$$M = \xi^* H^*_{e_i}\xi^{*\prime} + e^* H^*_{\xi_i}\xi^{*\prime} \tag{56c}$$

$$N = \xi^* H^*_{e_i} + e^* H^*_{\xi_i}. \tag{56d}$$

Since boundedness of (e^*, ξ^*) and stability of $(H^*_{e_i}, H^*_{\xi_i})$ are established by definition of the tuned system, it is not difficult to see that conditions for the stability of F and the boundedness of $\tilde{x}_L$ are identical. In fact, this follows if and only if the system $S:(x_0, W)| \to x$, described by

$$\dot{x} = A_0(w - Mx),\ x(0) = x_0 \in R^n \tag{57}$$

is stable (Kosut, 1983). Note that the system S is identical in form to the linearized parameter error system $(\tilde{\pi}_0, \xi^* e^*)| \to \tilde{\pi}_L$ in (54).

Assuming that S is L_∞-stable we obtain the following local stability

Theorem 2

Suppose $F \in L_\infty$-stable and $\tilde{x}_l \in L_\infty$. Hence, there exists a constant c such that

$$\gamma_\infty(F) \le c < \infty \tag{58}$$

Under these conditions, if, for some $\varepsilon < 2/c$,

$$\|\tilde{x}_l\|_\infty \le (1 - \varepsilon c/2)\varepsilon \tag{59}$$

then

$$\|\tilde{x}_l\|_\infty \le \varepsilon \tag{60}$$

Proof. The proof is entirely analogous to the proof of the linearization theorem on p. 131 of Desoer and Vidyasagar (1975). Details for this case may be found in Kosut (1983).

Discussion

Theorem 2 asserts that the adaptive system is stable, i.e. bounded inside an ε-region, provided that $F \in L_\infty$-stable and the linearized response is bounded and sufficiently small, i.e. condition (59). *No claims are made about the mechanism that provides $F \in L_\infty$-stable and $\tilde{x}_L \in L_\infty$.* As mentioned earlier, these are insured if the map S defined in (57) is L_∞-stable. It is possible, of course, that $\tilde{x}_L \in L^n_\infty$ but $\|\tilde{x}_L\|_\infty$ exceeds the magnitude constraint of (59). Instability, however, does not follow because Theorem 2 only provides sufficient conditions.

In order for theorem 2 to be of practical use, it is necessary to provide stability of S without relying on passivity of $H^*_{e_i}$. We will illustrate this by using persistent excitation. Consider the system

$$\dot{x} = -AfHf'x + u,\ x(0) \in R^n. \tag{61}$$

It is shown in Anderson (1977) that if $A \in R^{n \times n}$, $A = A' > 0$, $f \in \mathrm{PE}$ and $H \in \mathrm{SPR}$ then (61) is exponentially stable, i.e. there exists constants m, $\lambda > 0$ such that

$$|x(t)| \le m\bar{e}^{\lambda t}|x(0)| + \int_0^t m e^{-\lambda(t-\tau)}|u(\tau)|d\tau. \tag{62}$$

We will apply (62) to provide stability of S as follows. The system S can be written as

$$\dot{x} = -A_0\xi H_{e_i}\xi' x + A_0 w - Qx \tag{63a}$$

where ξ, H_{ci} and Q are defined via

$$H^*_{ci} = H_{ci} + \tilde{H}, \tag{63b}$$

$$\xi^* = \xi + \tilde{\xi} \tag{63c}$$

$$Q = A_0(M - \xi H_{ci}\xi) \tag{63d}$$

Comparing (63a) with (61), intuitively, if $\xi \in$ PE, $H_{ci} \in$ SPR, and Q sufficiently 'small' then the system (63) (equivalently the map S) remains exponentially stable. Thus, by Theorem 2, an ε-region of local stability exists. The precise conditions are stated as follows.

Corollary 2.1

Let $\bar{H}_{ci} \in$ SPR and $\bar{\xi} \in$ PE with corresponding positive constants λ and m as defined in (62). Then, $F \in L_\infty$-stable and $\tilde{x}_L \in L^n_\infty$ if

$$\lambda m > q = \|\bar{\xi}\|_\infty \|\bar{\bar{\xi}}\|_\infty (2 + \|\bar{\xi}\|_\infty)\gamma_\infty(H^*_{ci}) + \|e^*\|_\infty \|\xi^*\|_\infty \gamma_\infty(H^*_{ci}) \tag{64}$$

and

$$\gamma_\infty(\bar{H}_{ci}, \bar{\xi}) < (\lambda m - q)\|\bar{\xi}\|^2_\infty . \tag{65}$$

Proof. Follows directly by application of Small Gain Theory (Zames, 1966) to (63). Details may be found in Kosut (193).

Discussion

Corollary 2.1 shows that persistent excitation is one mechanism which provides $S \in L_\infty$-stable, and hence, boundedness of $\tilde{x}_L$ and stability of F. Therefore, if in addition, $\tilde{x}_L$ is sufficiently small (59) then the adaptive system has a local stability.

Other mechanisms to provide stability of S include dead-zones, retardation functions, and signal normalizations. Their effect on S needs to be determined.

Corollary 2.1 also provides an upper bound on the effect of model error via (65). This is not yet in the frequency domain form we would like, but the bound can be quite large. Hence, H^*_{ci} *need not be* SPR, but only approximately so, e.g. $\bar{H}^*_{ci} \in$ SPR. Think of H^*_{ci} being SPR only at low frequencies. In the same way, the signal $\bar{\xi}$ can be viewed as the dominant part of ξ^* causing excitation in that part of the spectrum where the model error is small, e.g. also at low frequencies. Ioannou and Kokotovic (1983b) also discuss this type of frequency separation in the regressor in the presence of high-frequency parasitics.

These results still remain incomplete because we need to know the relationships between λ, m and the 'size' of $\bar{\xi}$, e.g. Theorem 2 requires a bound on $\|\tilde{\pi}_J\|_\infty$, which is a function of λ, m and consequently $\bar{\xi}$. Of further interest is the effect of dead-zones and signal normalizations on the variational form (53). Certainly the nature of the memoryless nonlinearity $f(x)$ changes, as well as the system S.

6. CONCLUSIONS

In this paper we have presented a framework for an input output theory of adaptive control. This viewpoint provides a means to realistically determine the robustness properties of adaptive algorithms. Moreover, input output concepts are closely related to measurement techniques, and hence, can lead to the determination of usable engineering techniques. In control design and analysis the most notable example is the use of Bode plots for scalar systems (Bode, 1945) and singular value plots for multivariable systems (Doyle and Stein, 1981). At the present time, no similar 'engineering theory' exists for adaptive control design. *En route* to establishing such a theory it will be necessary to resolve some of the open issues raised herein. The possible benefit to adaptive control engineering design is substantial.

Acknowledgements R.L.K. was partially supported by Air Force Office of Scientific Research (AFOSR) Contract F49620-83-C-0107. C.R.J. was supported by National Science Foundation Grant ECS-8119312.

REFERENCES

Anderson, B. D. O. (1977). Exponential stability of linear equations arising in adaptive identification. *IEEE Trans. Aut. Control*, **AC-22**, 83–88.

Anderson, B. D. O. and C. R. Johnson, Jr (1982a). Exponential convergence of adaptive identification and control algorithms. *Automatica*, **18**, 1–13.

Anderson, B. D. O. and C. R. Johnson, Jr (1982b). On reduced order adaptive output error identification and adaptive IIR filtering. *IEEE Trans. Aut. Control*, **AC-27**, 927–933.

Anderson, B. D. O. and R. M. Johnstone (1983). Adaptive systems and time varying plants. *Int. J. Control*, **37**, 367–377.

Aström, K. (1983). Theory and applications of adaptive control—a survey. *Automatica*, **19**, 471–486.

Bode, H. W. (1945). *Network Analysis and Feedback Amplifier Design*. Van Nostrand, Princeton, NJ.

Desoer, C. A. and M. Vidyasagar (1975). *Feedback Systems: Input–Output Properties*. Academic Press, New York.

Doyle, J. C. and G. Stein (1981). Multivariable feedback design: concepts for a modern classical synthesis. *IEEE Trans. Aut. Control*, **AC-26**, 4–17.

Egardt, B. (1979). *Stability of Adaptive Controllers*. Springer-Verlag, Berlin.

Gawthrop, P. J. and K. W. Lim (1982). Robustness of self-tuning controllers. *IEE Proc.*, **129**, 21–29.

Gomart, O. and P. Caines (1984). Robust adaptive control of time-varying systems. McGill University Research Report, Department of Electrical Engineering.

Goodwin, G. C., P. J. Ramadge and P. E. Caines (1980). Discrete time multivariable adaptive control. *IEEE Trans. Aut. Control*, **AC-25**, 449–456.

Ioannou, P. A. and P. V. Kokotovic (1983a). Improvement of robustness of adaptive schemes. *Proceedings of the 3rd Yale Workshop on Applications of Adaptive Systems Theory*, June 1983, pp. 148–153.

Ioannou, P. A. and P. V. Kokotovic (1983b). *Adaptive Systems with Reduced Models*. Springer-Verlag, Berlin.

Kosut, R. L. (1983). Robust adaptive control: conditions for local stability. *Proceedings of the 1983 Automatic Control Conference*, San Francisco, CA, June 1983.

Kosut R L and B Friedlander (1982) Performance robustness properties of adaptive control systems *Proceedings of the 21st IEEE Conference on Decision and Control* Orlando, FL December 1982 pp 18 23

Kosut R L and B Friedlander (1983) Robust adaptive control conditions for global stability *IEEE Trans Aut Control*

Kosut R L , C R Johnson Jr and B D O Anderson (1983) Conditions for local stability and robustness of adaptive control systems *Proceedings of the 22nd IEEE Conference on Decision and Control* San Antonio, TX, December 1983, pp 972 976

Kreisselmeier G and K S Narendra (1982) Stable model reference adaptive control in the presence of bounded disturbances *IEEE Trans Auto Control* **AC-27**, 1169 1175

Landau Y D (1978) Elimination of the real positivity condition in the design of parallel MRAS *IEEE Trans Aut Control*, **AC-23** 1015 1020

Monopoli R V (1974) Model reference adaptive control with an augmented error signal *IEEE Trans Aut Control*, **AC-19**, 474 484

Narendra, K S , Y H Lin and L Valavani (1980) Stable adaptive control design *IEEE Trans Aut Control*, **AC-25**, 440 448

Ortega, R and I D Landau (1983) On the model mismatch tolerance of various parameter adaptation algorithms in direct control schemes A sectoricity approach *Proceedings of the IFAC Workshop on Adaptive Control* San Francisco CA June 1983

Praly L (1983) Robustness of indirect adaptive control based on pole placement design *Proceedings of the IFAC Workshop on Adaptive Control*, San Francisco, CA June 1983

Rohrs C E , L Valavani, M Athans and G Stein (1981) Analytical verification of undesirable properties of direct model reference adaptive control algorithms *Proceedings of the 20th IEEE Conference on Decision and Control* San Diego CA, December 1981, pp 1272 1284

Rohrs, C E , L Valavani M Athans and G Stein (1982) Robustness of adaptive control algorithms in the presence of unmodeled dynamics *Proceedings of the 21st IEEE Conference on Decision and Control*, Orlando, FL, December 1982

Safonov, M G (1980) *Stability and Robustness of Multivariable Systems* MIT Press Cambridge Mass 1980

Samson, C (1983) Stability analysis of adaptively controlled systems subject to bounded disturbances *Automatica* **19** 81 86

Zames G (1966) On the input output stability of time-varying nonlinear feedback systems *IEEE Trans Aut Control* Part I **AC-11**, 228 238, Part II **AC-11** 465 476

Zames, G and B A Francis (1983) Feedback minimax sensitivity, and optimal robustness *IEEE Trans Aut Control* **AC-28**, 586 601

APPENDIX B

Multivariable Adaptive Control Algorithms and Their Mechanizations for Aerospace Applications

MULTIVARIABLE ADAPTIVE CONTROL ALGORITHMS AND THEIR MECHANIZATIONS FOR AEROSPACE APPLICATIONS

by

Sunil C. Shah, Robert A. Walker, and Hossein A. Saberi

Integrated Systems, Inc.
101 University Avenue
Palo Alto, CA 94301

I. INTRODUCTION

Aerospace control problems pose challenging requirements which recent advances in adaptive control are beginning to address. Several example applications and a discussion of aerospace applications in general, contrast with previous industrial and process control problems and illustrate the class of problems of interest. Because of the stringent performance requirements, mechanization issues strongly influence algorithm design. The freedom possible in adaptive algorithm design is outlined below with particular emphasis on computational costs for various algorithms and model forms. In particular the advantages of the lattice form model structure are pointed out below, with a description of a new formulation for the application of lattice forms to the control problem.

Parallel processing, while significant for many applications is not a panacea for the control problem. The potential drawbacks and the resolution of this important issue are addressed in Section IV. The implications of the nature of aerospace problems amenable to adaptive control, and the state of adaptive control research suggest a set of capabilities and tools which will greatly facilitate the design process. The characteristics of these capabilities and tools and how they should be realized in hardware and software are discussed in Section V. Section VI summarizes the major conclusions on the state and potential for adaptive control application to aerospace applications.

II. AEROSPACE ADAPTIVE CONTROL PROBLEMS

Several aerospace problems with demanding requirements have been addressed recently with adaptive control techniques. Examples are:

1. rotorcraft vibration suppression [1, 2]
2. large space structure pointing control [3], and
3. aircraft wing/store flutter suppression [4].

Each of these problems have differing characteristics which are summarized in Table I below.

General characteristics present in most aerospace applications include:

1) High bandwidths and consequently a high sampling rate requirement - rates of 200 to 2000 hertz are not uncommon.

2) Most of the problems are multivariable,

3) Often a priori information, based on physical models of the vehicle dynamics are quite good (certainly better than the typical process control situation). Even where physical models do not exist, often good empirical models can be developed by ground vibration, wind tunnel, or flight tests.

4) Weight and power constraints put a higher premium on computational cost, in contrast to say a ship autopilot or industrial controller,

5) Aerospace mission requirements generally require more comprehensive prototyping and testing to reduce full system test time and ensure high operational reliability.

TABLE I [illegible]

Problem Characteristics	ROTORCRAFT VIBRATION SUPPRESSION	LARGE SPACE STRUCTURE POINTING CONTROL	AIRCRAFT WING/STORE FLUTTER SUPPRESSION
Nature of Adaptive Requirement	• Fixed narrowband modes for different flight and operating conditions, for different flexible rotorcraft	• Ground Testing impossible as well as many modes that changes with temperature and aging	Level 1 • Adjust to differing flight conditions Level 2 • Adapt to sudden model instabilities from unanalyzed store configurations
Key a prior information	• Disturbance frequency	• Sufficient time to partition and identify model bandwidth	• Timing of when stores are dropped
Special Identification Aspects	Near singular disturbance that does not enter through the control	• Multiple narrow and wideband disturbances	• Very short time to identify • Model mismatch and hence model form are very important • Physical model tuning parameters are very helpful
Special Control Problem Aspects	• Inherently MIMO • Often nonminimum phase • Autotuning capability is essential	• Need very high disturbance rejection gain • Controller bandwidth is packed with modes • Off-line robustness analysis and design are helpful	• Stability is the key problem • Some off-line analysis of a priori models and stability boundaries essential to guarantee stability
Adaptive Mechanization Aspects	• Frequency-shaped time domain model concentrates controller energy • Notch filter reduces parameter estimation load • Hysteresis adaptive on/off logic bypasses persistent excitation problems	• Decentralized control may be valuable due to model complexity, system reliability, computational requirements and stepwise deployment	• High computational load and therefore can benefit from parallel architectures • Persistent excitation is a problem: turning on/off excitation, freezing adaptation, and covariance modification are potential solution

III. COMPUTATIONAL ASPECTS OF ADAPTIVE CONTROL

Throughout this paper, we will denote the state dimension of the model by n and the number of outputs by p. We will restrict our attention to the case of equal inputs and outputs.

Typical adaptive control schemes can be conceptually divided in three parts:

(1) Parameter estimation ($O(n^2p)$ for RLS type update $O(np)$ for lattice

(2) Control design ($O(1)$ for direct adaptive control, $O(n^3p^3)$ for pole-placement

(3) Control update ($O(np)$ for vector ARMAX, $O((n+p)^2)$ *for a state space model*).

The second step may be trivial for direct adaptive control. The first two steps are usually computationally the most expensive.

Figure 1 below gives the computation cost of step (1) and (2) as a function of state dimension of the identified model, for a 2-input 2-output system using a U-D factored covariance update and the minimum variance control strategy for the self-tuning regulator of Astrom and Wittenmark.

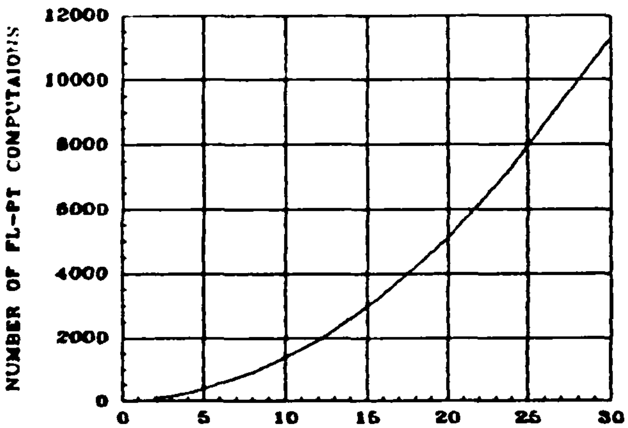


Figure 1. *Self-tuning Regulator with U-D Update*

Assumptions:

(1) 1 flop = 1 floating point add + 1 floating point mult
= 2 floating point adds

(2) The vector ARMAX model has uniform delay=1 from input to output. The observability indices of the corresponding state-space model are generic. i.e., = n/p, (See Hannan [5] on generic parameterizations).

Figure 2 below gives the computational cost of implementing the servomechanism form of a one-step ahead minimum variance controller as a function of state dimension. Other assumptions are the same as for Figure 1.

One obvious conclusion from these cost estimates is that for a machine with floating point speed of about 200k flops per second (about that of a VAX 11-780 with floating point hardware for single precision arithmetic according to Dongarra [6]), the update rate for a 10 state model can not exceed 150 Hz and 70 Hz for the two cases above.

Figure 3 below shows the achievable rate for a machine capable of performing 200k flops per second as a function of state dimension for a 2-input 2-output system. By comparison the control update in step (3) takes only 4n and 6n flops for the regulator and controller cases mentioned above, and hence is negligible.

Typical aerospace applications need control update rates of [illegible] 200 Hz and 2 kHz so that it is

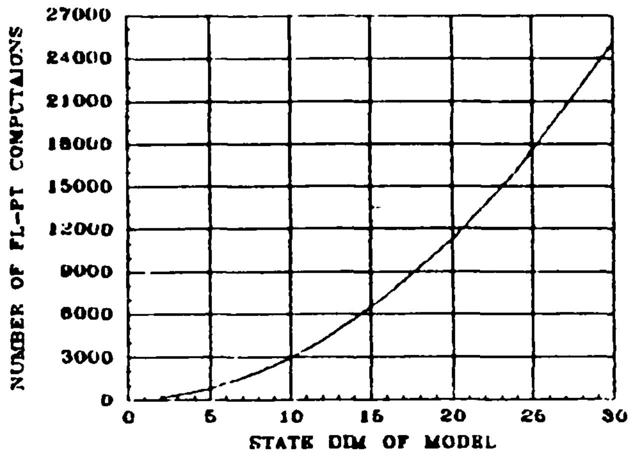


Figure 2. Self-tuning Controller with U-D Update

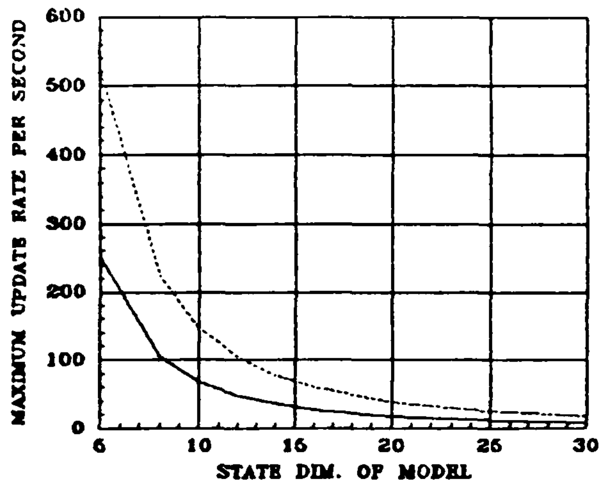


Figure 3. Achievable Parameter Update Rates on a VAX 11/780 Class Machine for 2-input 2-output Minimum Variance Controller in Single Precision Arithmetic. Solid Line is for the Controller Case and the Broken Line is for the Regulator Case

clear that one cannot expect to update parameters at the control update rate, at least with a typical 200k flop serial machine. Most flight-qualified machines available today are rated at much lower speeds compared to the VAX 11-780. Consequently it is necessary to consider algorithms that use slower update rates for parameters, compared to the signal sampling rate. It is also necessary to consider parallel hardware for adaptive control algorithms.

Separation of Update Rates For Parameter Estimation and Control.

We consider several options for running parameter estimation at a slower rate R/k than the control update rate R. We assume that the data is available at a rate corresponding to the control update.

(1) Skip every k-1 points of input output data and use the kth point for parameter estimation. Perform rate change from the identified model to the control update rate model

[illegible] point. Perform the control update based on this model at the full rate.

(2) Perform the identification update every kth [illegible] the regressors of the model at the [illegible] rate h. The control and identification models are at the same rate, enabling direct algorithms to be used.

The second alternative has two advantages over the first one. The computation cost of performing change of rate is avoided, but more importantly the first method will result in k fold reduction in the Nyquist rate with the consequent aliasing and loss of identifiability. Analysis of the second alternative for the situation where identifiability is guaranteed by proper parameterization and persistency of excitation can be done readily. Convergence of the parameters can be established (provided the SPR condition [7] or the sector condition, see Kosut [8], is met) by a simple modification of the standard proofs in Goodwin and Sin [9]. Analysis of the first alternative has not been done yet.

IV. PARALLEL ARCHITECTURES FOR ADAPTIVE CONTROL

It is possible to increase the update rate for identification by using parallel architectures and/or different algorithms. In particular the lattice forms algorithms can provide very fast parameter update rates for high order systems. Recently Jover and Kailath [10] have shown that measurement update can be implemented in a parallel pipelined form.

Lattice Wavefront Architecture

The basic lattice form algorithm simply involves an efficient mechanism for obtaining the least-squares estimates of a linear-in-parameter model.

The identification update uses the model

$$\hat{y}_{t+1} = \hat{\theta}_t \varphi_t$$

and the least squares update simply produces the parameter $\hat{\theta}_N$ such that $\frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2$ is minimized. The steps required for the lattice algorithm are described in Table II. The main advantages of the lattice form parameter update are

(1) O(np) computation cost. However, for low parameter dimensions (n<16), U-D is still more cost effective.

(2) Simultaneous lower order model identification.

(3) Increase in model order (2p states per time point) is straightforward.

(4) Suitable for modular wavefront array processing architectures (Lev-Ari [11]).

(5) It is possible to perform most computations in fixed point arithmetic.

Lattice form algorithms have for some time not been considered appropriate for adaptive control because of the perceived need to convert the lattice form structure to the ARMAX structure so that the familiar controller design techniques could be used. However, direct adaptive control laws such as the minimum variance type control can be implemented directly with the lattice form model (Shah [12]). The [illegible] is to use trial inputs u_t in the fixed [illegible] lattice prediction model to generate the u_t that will make the prediction $\hat{y}_{t+1} = 0$ based on the current parameter estimate.

When a wavefront architecture is used for implementation of the lattice update, speed up proportional to the numbers of processors can be achieved, provided delayed least-squares parameter estimates can be used in the controller.

It is straightforward to analyze the serial hardware case of adaptive control by reference to results on the equivalent least-squares. The parallel architecture algorithm with its attendant delays in the coefficients due to pipelining has not been analyzed rigorously yet. It appears that if identifiability of the parameters is ensured by model structure selection and persistency of excitation, then by invoking quasi-stationarity of the parameter estimates one could show stability and convergence.

TABLE II. ADAPTIVE CONTROL ALGORITHM USING LATTICE FORM:

True System: $A(q^{-1})y_{t+1} = B(q^{-1})u_t + (q^{-1})V_t$ Vector (ARMAX)

Identification Model:

Main lattice uses past values of

$$x_t = \begin{bmatrix} u_t \\ y_t \\ y_{ref_{t+1}} \end{bmatrix}$$

Joint lattice predicts $\hat{y}_{t+1} | x_t, x_{t-1}, x_{t-m}$.

Identification Method

The lattice coefficients $\hat{\theta}_\ell$ minimize

$$\sum_{t=0}^{N} (y_t - \hat{y}_{t+1} | x_t, x_{t-1}, x_{t-m})^2 \lambda(N-t), \; x_t = 0 \text{ for } t < 0.$$

Prewindowed form, multichannel, cellular.

Control Computations:

- Truncate the model order if indicated by low residuals power.
- Find the linear affine map $u_t \mapsto \hat{y}_{t+1}$ for the lattice filter with the identified $\hat{\theta}_t$. Invert the map to obtain u_t such that $y_{t+1} = y_{ref_{t+1}}$. This is simply linear solution of p equations in p unknowns.

This algorithm of (1) finding a linear least squares estimate in a linear regression model and (2) then computing the control to make the prediction based on the current parameter estimates equal to some known value, lies at the heart of a large number of "successful" adaptive control algorithms. The same basic algorithm is also used for recursive prediction algorithms.

Convergence and stability analysis of these algorithms has been done extensively [13-17]. Many practical applications have also been reported for this class of algorithms. It is known that this class of algorithm requires a Strict Positive Real (SPR) condition, i.e., [C (q) -] has to be SPR for convergence.

To obtain the lattice form algorithm, an earlier attempt by Friedlander [18] involved a joint process estimation of the ARMAX system. The joint estimation form involved the feedback part of the system and the implementation required computing the polynomial coef-

... iteration and enforcing the known ... structure on the estimates. No convergence analysis could be performed and the numerical burden remained $O(n^2)$.

Parallel U-D Factored Measurement Update

The main disadvantage of the lattice form algorithm is that is it computationally costly to convert lattice form models to their polynomial forms. It is very hard to convert any a priori information, which is generally in the state-space form, to the lattice form.

The U-D factored measurement update Kalman filter formulation has the best potential for incorporating a priori information in the state-space form. Jover and Kailath [17] describe a parallel architecture to implement the measurement update so that with 2n+4 processors, each performing one floating point addition one floating point multiply and data transfer to and from its neighbors in T units, can process a new p dimensional measurement every p(n+1)T units. (n is the state dimension for the Kalman filter). The input-output delay between new measurements and the corresponding new state estimates is (2n+6)T units.

For the adaptive control algorithms the state estimates represent the parameter estimates and consequently only delayed parameter estimates are available for control computations. Computation of the sensitivity matrix $\frac{\partial y}{\partial \theta}$ in the Extended Kalman Filter Formulation applied to state-space models (Ljung [19]) can be very costly. Generally only local convergence results can be established in such cases.

V. TOOLS FOR MECHANIZATION OF ADAPTIVE CONTROL

Proper tools can expedite mechanizations of adaptive control algorithms tremendously. A representative toolkit for various phases of adaptive control mechanization is described here. Table 3 below summarizes these tools and how they are used.

Such a simulation and analysis toolkit is being developed in MATRIX$_x^{TM}$/System_Build. This toolkit involves two distinct environments -- an analysis and simulation environment and a real-time implementation environment. Both environments share the same model definition data structure. The top three blocks in Table 3 relate to analysis/simulation, while the bottom two are referring to real-time adaptive control.

Figures 4.a - 4.c below show an example adaptively controlled system specified in graphical block diagram form. Simulations of the candidate algorithm are performed before transferring the control structure to a real-time processor.

VI. CONCLUDING REMARKS

The bandwidth, system order, MIMO nature, and often non-minimum phase characteristics of aerospace ... performance applications make them an excellent stimulus to a much more complete working theory of adaptive control. What will speed the implementation of adaptive control laws in such an environment is a set of tools for interactive specification of the adaptive control laws, simulation and analysis of their performance, and efficient transfer of the control law to a real-time processor.

TABLE 3. ADAPTIVE CONTROL MECHANIZATION PROCESS AND TOOLS

Functional step	Tool
Model building incorporating a priori physical information. Adaptive algorithm specification	Interactive graphic specification of model [illegible] fig 4 below
Parameter and structure estimation signal processing	[illegible] data acquisition [illegible] identification and signal processing
Simulation of discrete-continuous system. Analysis of various parts	System simulator and [illegible]. [illegible] MIMO [illegible] Linearization, ODE analysis
Testing the adaptive algorithms on prototype hardware	High speed real time control hardware capable of implementing the algorithms used in simulations
Flight tests	Flight qualified version of the above hardware

same data struct

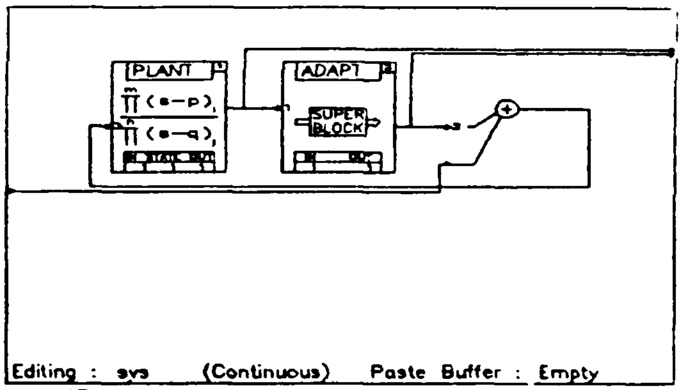


Figure 4.a. System_Build Block Diagram of an Adaptively Controlled Plant. (The Plant is Represented by s-domain pole-7160 pairs. The Discrete-time Adaptive Control along with the Samplers and ZOH are implemented in the SUPERBLOCK ADAPT on the right.)

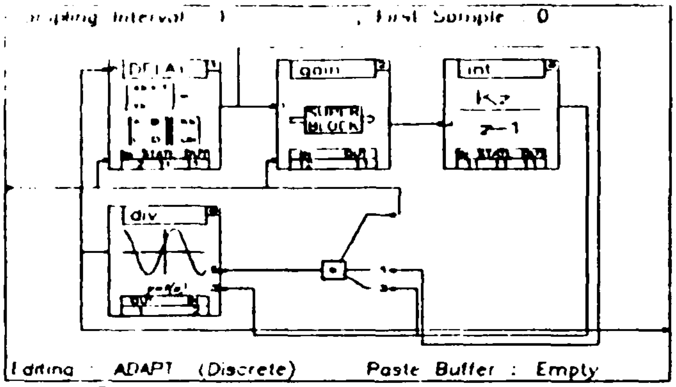


Figure 4. b. Discrete-time Adaptive Control Algorithm with Dead-zone (see Egardt [20])

Delay: Generates the Regressor Vector from y_t and u_t

Gain: Computes the gain for Parameter Update Using y_t and the Regressor Vector.

Int: Updates the Parameter Vectors using the Gain Estimate

Div: Computes the New Control using Parameter and the Regressor Elements.

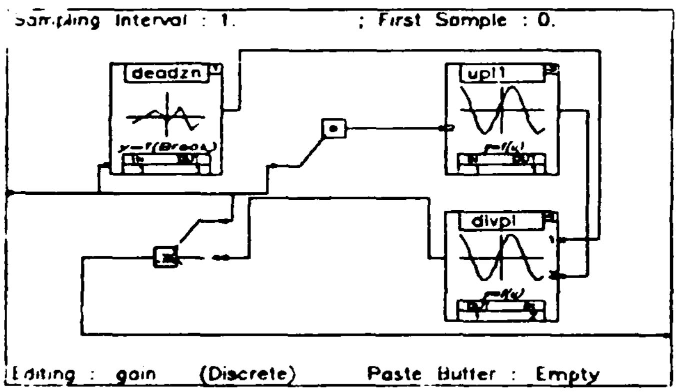


Figure 4.c. Parameter Estimate Gain Computations using a Deadzone (DEADZN in the Diagram) - The Top Centre Block Computes the Norm of the Regressors. The Remaining Elements Implement the Gain = $\frac{g_{t-1}\varphi_{t-1}y_t}{1+|\varphi_{t-1}|^2}$ which is the output of the SUPERBLOCK GAIN.

ACKNOWLEDGEMENTS

The authors would like to acknowledge U.S. Army support under Contract DAAK10-83-C-0188 and NASA support under Contract NAS2-11548, as well as thank Dr. Hanock Lev-Ari for helpful discussions.

REFERENCES

[1] W. Johnson, "Self-Tuning Regulators for Multicyclic Control of Helicopter Vibration," NASA Technical Paper 1996, March 1982.

[2] R. H. Roy, H. A. Saberi, R. A. Walker, "Rotorcraft Vibration Control: Time-Domain Model Based Fixed-Gain and Adaptive Techniques", ISI Report 45, under NASA Contract NAS2-11548, Integrated Systems, Inc., Palo Alto, Calif., August 1984.

[3] M. G. Lyons and R. L. Kosut, "Adaptive Techniques for Large Space Structures", ISI Report 43, under AFOSR contract F-49620-83-C-0107, July 1982.

[4] C. Hwang, E. H. Johnson, D. S. Joshi, "Test Demonstration of Digital Adaptive Control of Wing/Store Flutter: Part II - Demonstration of Adaptive Control," AFWAL-TR-82-3094 Part II, Dec. 1982.

[5] Hannan, E. J. "Developments in Statistics, Vol. II, Chapter 2, Academic Press: New York, NY, 1979.

[6] J. J. Dongarra, "Redesigning Linear Algebra Algorithms," Proc. of the International Collequim on Vector/Parallel Computing in Scientific Applications Conference, Paris, France, March 17-18, 1983.

[7] Solo, V. "The Convergence of AML," IEEE Trans. Auto. Control, Vol. AC-24, No. 6. pp. 958-962, 1978.

[8] Kosut, R. L., Johnson, R. C., "An Input-Output Vices of Robustness in Adaptive Control", Special Issue on Adaptive Control, Automatica, Sept. 1984.

[9] Goodwin, G. C. and Sin. K. S. "Adaptive Filtering, Prediction and Control." Chapter 8, Prentice Hall. Englewood, NJ., 1984.

[10] Jover, J. and Kailath, T., "A Parallel Architecture for Kalman Filter Measurement Update", Info. Syst. Lab. Report, Stanford University, Stanford, CA 1984.

[11] Lev-Ari, H. A. "Modular Architecture for Adaptive Multichannel Lattice Algorithms". IEEE Int. Conf. Acoust. Speech and Signal Proc. Apr. 1983.

[12] Shah, S. C. "Adaptive Prediction and Control Using Lattice Forms." Technical Report, Integrated Systems, Inc., ISI-10. 1981.

[13] Ljung, L. "On Positive Real Functions and Convergence of Some Recursive Schemes," IEEE Trans. Auto. Control, Vol. AC-22. pp. 539-551 (1977).

[14] Solo, V., "Time Series Recursions and Stochastic Approximation", PhD Thesis, Australian National University, Camberra, Australia (1978).

[15] Kumar, R., "Almost Sure Convergence of Adapative Identification, Prediction and Control Algorithms" Tech. Report. LCDS 81-8 Div. of Applied Math. Brown Univ., Providence, R. I. (1981).

[16] Sin, K. S. and Goodwin, G. C., "Stochastic Adaptive Control Using Modified Least Squares Algorithms." Automatica Vol. 18, No. 3, pp. 315-321, (1982)

[17] Ljung, L. and Sodestrom, T. "Theory and Practise of Recursive Identification" MIT Press, Cambridge, Mass. 1984.

[18] Friedlander, B. "Recursive Lattice Forms for Adaptive Control", JACC, San Frans. 1982.

[illegible] Ljung, L. "Asymptotic Behavior of the Extended Kalman Filter as Parameter Estimator for Linear Systems". IEEE Trans. Auto. Contr., Vol. AC-24, No. 1, pp. 36-51, 1979.

[illegible] Egardt, B., "Stability Analysis of Some Discrete-time Adaptive Control Systems," IEEE Trans. Auto. Control, Vol. AC-[illegible], No. 4, pp. 710-717.

APPENDIX C

NONLINEAR CONTROL DESIGN BY GAIN SCHEDULING

APPENDIX C

NONLINEAR CONTROL DESIGN BY GAIN SCHEDULING

Since aircraft systems are predominantly nonlinear, it is necessary to be precise about normally accepted notions used for linear systems such as: measures of signal magnitude (norms), gain, stability, etc. The relevant mathematics and notation which we will use throughout, can be found in [Des.1], [Vid 1].

2.1 INTRODUCTION

A highly successful approach to flight control design is gain scheduling. In fact, this approach can be applied to any nonlinear control problem A typical gain scheduled control system is shown in Fig 2-1, where $P:L_e^m \to L_e^m$ is the plant (aircraft), $d \in L^m$ is an output disturbance and $C:L_e^m \to L_e^m$ is the gain scheduled controller, whose gains are scheduled as a function of the actual trajectories (u,y). Note that C maps output error signals $\tilde{y} := \bar{y} - y$ into control error signals $\tilde{u} = u - \bar{u}$.

The first step in obtaining C is to design a collection of linear controllers based on linear models, each corresponding to a particular reference trajectory $(\bar{u}\ y)$ or flight/power condition in the flight regime. Thus, the resulting controller gains are a function of the reference flight/power condition $(\bar{u},\bar{y})$ at that point. The control gains are then 'scheduled' as functions of the actual flight/power condition (u,y) to achieve a continuous nonlinear control throughout the operating envelope.

Each reference $(\bar{u}_i\ \bar{y}_i)$ generates a linear perturbation model, denoted by P_{L_i}. This model can be obtained as a first order perturbation of P_{NL} i.e.,

$$P_{NL}(\bar{u}_i - \tilde{u}) = P_{NL}\bar{u}_i + P_{L_i}\tilde{u} + O(||\tilde{u}||^2) \qquad (2.2)$$

Repeating for k selected flight conditions $(\bar{u}_i\ \bar{y}_i)$, $i = 1, \ldots, k$ yields a corresponding collection of linear models $\{P_{L_1}, \ldots, P_{L_k}\}$. In the case when $(\bar{u}_i\ \bar{y}_i)$ is an equilibrium then $\bar{u}_i$ and $\bar{y}_i$ are constants and

P_{L_i} is LTI. When $(\bar{u}_i,\bar{y}_i)$ is a dynamic trajectory, then P_{L_i} is LTV (linear time varying). Usually only equilibria are selected, however, to fully account for the behavior of high performance aircraft, it is necessary to consider dynamic references as well as equilibrium references.

Having determined a collection of linear models $\{P_{L_1}, \ldots, P_{L_k}\}$ corresponding to the k-nominal trajectories $\{(\bar{u}_1,\bar{y}_1),\ldots,(\bar{u}_k,\bar{y}_k)\}$, any number of design techniques can be used to determine a set of linear controllers $\{C_{L_1},\ldots,C_{L_k}\}$. The ith linear controller C_{L_i} is indirectly a function of the ith trajectory $(\bar{u}_i\ y_i)$. Connecting this collection of controllers as a function of the actual $(u\ y)$ is 'gain scheduling.' The resulting controller C is nonlinear. The same scheduling procedure can be used to connect the collection of linear models as shown in Fig 2-2. The resulting nonlinear model is often referred to as a 'simplified nonlinear model,' denoted by P_{SNL}.

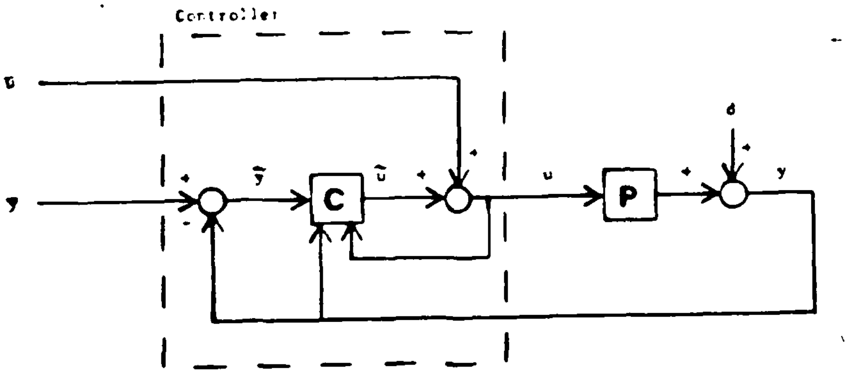


Figure 2-1. Gain-Scheduled Control System

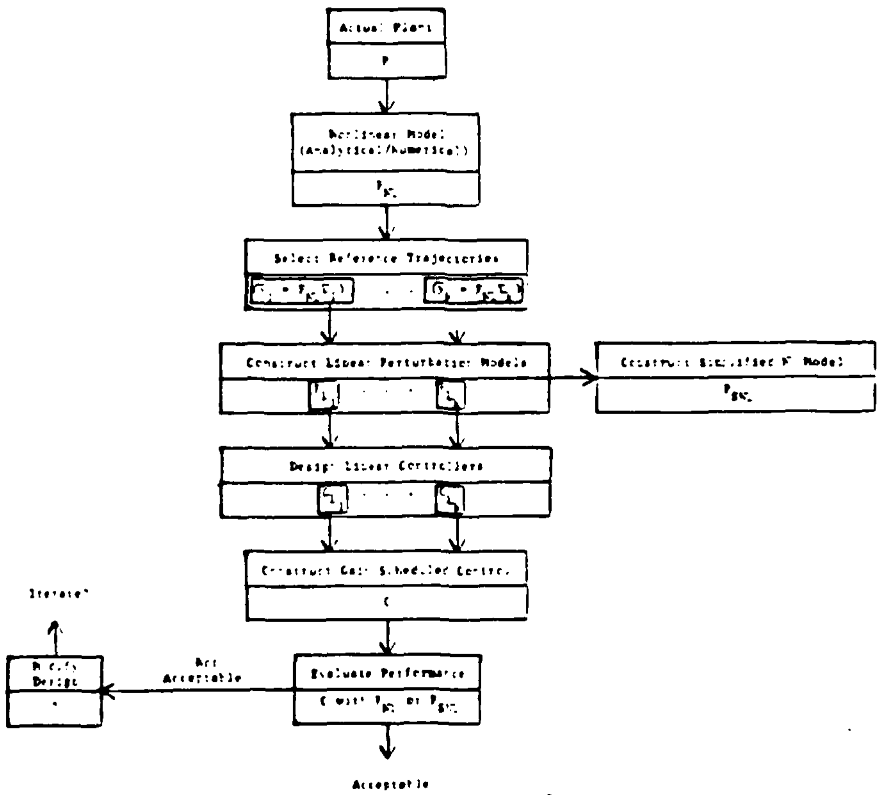


Figure 2-2. Gain Scheduling Design Procedure

Limitations in Theory

A fundamental issue in the gain scheduling procedure is that there is no theoretical justification that the resulting nonlinear (gain scheduled) control will provide acceptable performance while the vehicle is in transit from one flight/power condition to another. The difficulty lies in the fact that the linear models are only known to be valid at specific conditions, and no analysis has been done to evaluate the effect of modeling error. Thus, in the evaluation phase (Fig. 2-2), if performance is not acceptable then it is not well understood how to modify the design.

In Section 2-4 we precisely define conditions under which the gain-scheduled controller will work throughout the flight envelope despite (not necessarily small) modeling error (see Theorem 1). A key element in this result is the quantification of modeling error, which is discussed in the next section.

2.2 MODEL ERROR

The effectiveness of the linear model P_{L_i} in the neighborhood of the ith reference $(\bar{u}_i, \bar{y}_i)$ can be evaluated from the test set-up shown in Fig 2-3.

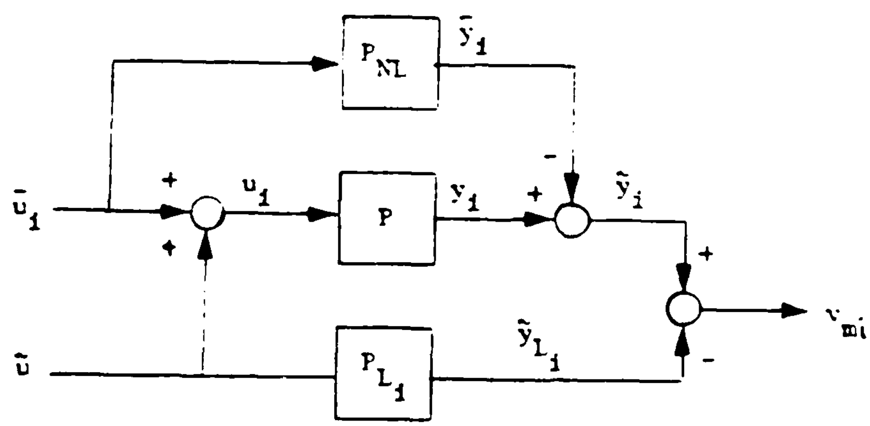


Figure 2-3. Model Error Test Set-Up

The model error test signal is defined as.

$$v_{mi} := P(\bar{u}_i + \tilde{u}) - P_{NL}\bar{u}_i \quad P_{L_i}\tilde{u} \tag{2.3}$$

Even when $\tilde{u} = 0$ there is an inherent model error bias,

$$b_{mi} := ||v_{mi}||_p \Big|_{\tilde{u}=0} = ||(P-P_{NL})\bar{u}_i||_p \tag{2.4}$$

The bias error is principally due to unmodeled dynamics. When $\tilde{u} \neq 0$, he model error will be larger. Since we ultimately seek a means to assure that perturbations $(\tilde{y}_i . u_i)$ remain in a bounded neighborhood of the reference, it is logical to construct a local model error test We introduce the following definitions of local gain.

Definition: An operator $G\, L^m_{pe} \dashrightarrow L^m_{pe}$ has finite local gain if $\forall$ W ε L_p stable and $Vb \geq 0$, $\varepsilon > 0$ such that:

$$||u||_p \leq \varepsilon \Rightarrow ||Gu||_p \leq ||Wu||_p + b \tag{2 5}$$

An operator G has finite local incremental gain if $\forall$ $\tilde{W}$ ε L_p-stable $\tilde{\varepsilon} > 0$ such that:

$$||u_1-u_2||_p \leq \tilde{\varepsilon} \Rightarrow ||Gu_1-Gu_2||_p \leq ||\tilde{W}(u_1\ u_2)||_p \tag{2 6}$$

It is convenient to introduce the nonlinear model error operator $\Delta_{mi}:L^m_{pe} \to L^m_{pe}$, defined implicitly by,

$$v_{mi} := \Delta_{mi}\tilde{y}_i \tag{2.7}$$

Let Δ_{mi} have finite local gain W_{mi} with bias b_{mi}, and finite local incremental gain $\tilde{W}_{mi}$. Although the bias b_{mi} can be directly calculated (2.4) it is not trivial to find a corresponding (ε, W_{mi}) and $(\tilde{\varepsilon}, \tilde{W}_{mi})$.

For example, consider LTI local gains $(W_{mi}, \tilde{W}_{mi})$ with transfer functions $W_{mi}(s)$, $\tilde{W}_{mi}(s)$. Let the norm be the L_2-norm This is convenient since the L_2-norm is connected to the frequency domain via Parseval's Theorem, i.e.,

$$||x||_2 = \left(\int_0^\infty |x(t)|^2 dt \right)^{1/2} = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |x(j\omega)|^2 d\omega \right)^{1/2} \qquad (2.8)$$

It is a simple matter to select a sinusoidal perturbation $\tilde{u}$ of frequency ω such that $_2 \tilde{y}_{i\,2} = \varepsilon$. Thus, W_{mi} can have any transfer function $W_{mi}(s)$ whose norm is found from the RMS test,

$$|W_{mi}(j\omega)| = [\mathrm{RMS}(v_{mi}) - b_{mi}]/\mathrm{RMS}(\tilde{y}_i) \qquad (2.9)$$

Similarly, for two inputs $\tilde{u}_1$ and $\tilde{u}_2$, we can find a bound on the incremental gain transfer function $\tilde{W}_{mi}(s)$.

Figure 2-4 shows modeling error vs. frequency for the linear model P_{L_1} associated with flight condition $(\bar{u}_1\ \bar{y}_1)$. Figure 2-5 shows the modeling error vs. frequency for flight condition $(\bar{u}_2, y_2)$. Note the increased modeling error when using the model P_{L_1} for a different flight condition. Also, the error in Figs. 2-5 and 2-6 remains substantially unaffected for frequencies above ω_m where the linear model is not valid

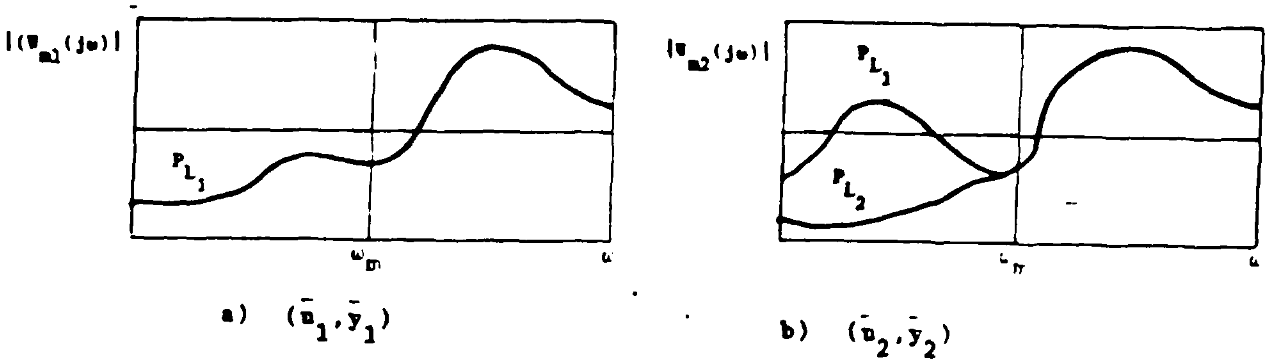


Figure 2-4. Model Error vs. Frequency

These tests can be repeated for each of the selected flight conditions to yield error curves like Fig. 2-4. Although these curves are readily obtainable from available test data, in practice they never are obtained, because it is not clear how to use them. The IMAC procedure, however, will exploit this data.

Limitations

Although this procedure works for LTI systems (see, e.g., [Kos.1]) it does not hold for NL systems, since sums of sinusoids at the input do not produce the same frequencies at the output. In other words, we do not know how to span the input/output space of an arbitrary nonlinear system. This is an area for further basic research.

Control Gain Error

The preceding discussion about model error can be repeated exactly for control gain error, defined as:

$$v_{gi} := C\tilde{u} - C_{L_i}\tilde{u} \tag{2.10}$$

where C is the gain scheduled controller and C_{L_i} is the linear controller corresponding to the ith reference $(\bar{u}_i, \bar{y}_i)$. The test set-up is shown in Fig. 2-5. By analogy with (2.7) introduce the nonlinear gain error operator $\Delta_{gi} : L^m_{pe} \rightarrow L^m_{pe}$, defined implicitly by

$$v_{gi} := \Delta_{gi}\tilde{u} \tag{2.11}$$

Let Δ_{gi} have finite local gain W_{gi} (with zero bias) and finite incremental gain $\bar{W}_{gi}$.

2.3 THEORETICAL BASIS FOR IMAC

The theoretical foundation for our IMAC methodology is to quantitatively assess the impact of model error and gain error on control objectives. Using the definitions of model error (2.7) and gain error (2.11), the gain-scheduled system (Fig. 2-4) can be put in the form shown in Fig. 2-6 where

$$e = e_L - Gv$$

$$v = \Delta e \tag{2.12}$$

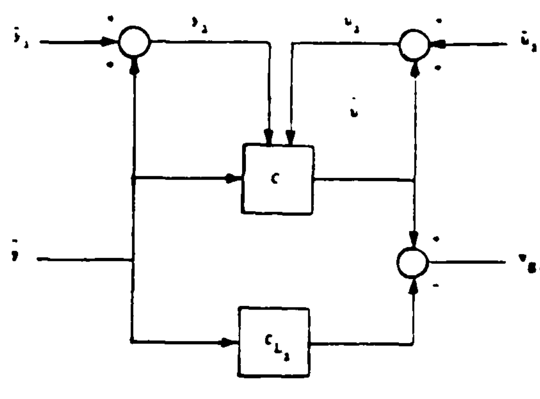


Figure 2-5. Gain Error Test

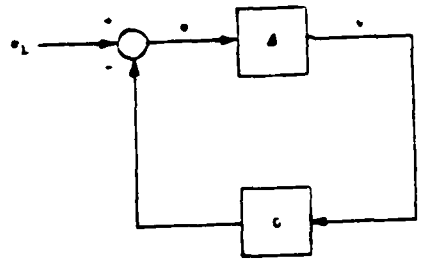


Figure 2-6. Equivalent Gain-Scheduled System

Results on stability and performance robustness are contained in Theorem 1, below.

<u>Theorem 1</u>:

Consider the system of Fig. 2-6, as described by (2 12). Assume that:

(A1) $e_L \in L_p^m$

(A2) $G \in L_p$-stable (G is also linear)

(A3) Δ has finite local gain, i.e., $\forall\, W \in L_p$ stable and $\forall b \geq 0$, $\varepsilon > 0$ such that:

$$||e||_p \leq \varepsilon \Rightarrow ||\Delta e||_p \leq ||We||_p + b$$

(A4) Δ has finite local incremental gain, i.e., $\forall\, \tilde{W} \in L_p$ stable, $\tilde{\varepsilon} > 0$ such that:

$$||e_1 - e_2||_p \leq \tilde{\varepsilon} \Rightarrow ||\Delta e_1 - \Delta e_2||_p \leq ||\tilde{W}(e_1 - e_2)||_p$$

Under these conditions, the system is (locally) L_p-stable if:

(i) $\gamma_p(\tilde{W}G) < 1$

(ii) $||e_L||_p \leq \tilde{\varepsilon} < \varepsilon$

(iii) $||e||_p \leq \varepsilon + \gamma_p(G)[1-\gamma_p(\tilde{W}G)]^{-1}[\gamma(W)\tilde{\varepsilon} + b] \leq \varepsilon$

Proof:

Application of the Linearization Theorem of [Des.1].

General Remarks

Assumptions (A1) and (A2) assert that the closed-loop perturbation (local) system is stable. This, of course, is established by the local design procedure

The remarkable aspect of Theorem 1 is that conditions (i) - (iii), which guarantee stability and performance, only involve linear operators, despite the fact that the actual system (Fig. 2-4) is nonlinear. The effects of the actual nonlinear system appear implicitly in the local gains/bias $(W, \tilde{W}, b)$. Conditions (i) - (iii) must be repeated along the significant equilibrium and dynamic references in the flight envelope.

Stability Robustness

Condition (i) is a generalization of the usual stability robustness test (see, e.g., [Doy.1], [Saf.1], [Kos.1]) but includes LTV systems as well as LTI systems.

Performance Robustness

Conditions (ii) (iii) must be satisfied to guarantee that $||\tilde{y}||_p \leq \varepsilon$, i.e., given model error $(W, \tilde{W})$ and performance tolerance ε, then it remains to select C_L such that (ii) - (iii) is satisfied. These conditions provide the means to quantitatively modify unacceptable performance (see flowchart in Fig. 2-5).

In the case where the actual system is LTI $(P, P_{NL} \varepsilon \underline{G})$, all the gain operators $\gamma_p(.)$ in (i) - (iii) can be replaced with the matrix norm $_{2\ 2}$ and (i) - (iii) become frequency dependent (see, e.g., [Kos 1] for details). This property is lost for the nonlinear problem at hand.

Selection of Norm Measure

The L_2 norm, which is useful in model error analysis and stability robustness, does not give a complete performance measure. For example, $||e||_2 \leq \alpha$ does not preclude $|e(t)| >> \alpha$ for some $t \in [0,\infty)$. However, $||e||_\infty \leq \beta$ together with $||e||_2 \leq \alpha$ gives a more complete performance evaluation. Conditions (i) (iii) can be used with the L_∞ norm but this impacts the model error (or gain error) test procedure, and opens an area for basic research in model error testing

2.4 MODEL ERROR ALLOCATION

Theorem 1 provides the means to examine the model error allocation problem: given performance bounds $(\varepsilon, \tilde{\varepsilon})$ find allowable model error $(W, \tilde{W}\ b)$. This is a difficult and important problem, since the solution allows for approximate model building and evaluation at an early stage of design.

Consider the simple case where local gain and local incremental gain are approximately identical, i.e.,

$$W \approx \tilde{W} \tag{2.10}$$

For sufficiently small $(\varepsilon,\tilde{\varepsilon})$ it is possible that W is close to $\tilde{W}$. However, even under these simple conditions we cannot isolate W in (2.14) without introducing undue conservatism. Specifically, although

$$\gamma_p(WG) \leq \gamma_p(W)\gamma_p(G) \tag{2.11}$$

it is possible that $\gamma_p(WG) << \gamma_p(W)\gamma_p(G)$, ergo, significant conservatism. Even so, (2.10), (2.11), together with (2.14) give,

$$\gamma_p(\tilde{W}) \approx \gamma_p(W) \leq \frac{\varepsilon\ \tilde{\varepsilon}}{\varepsilon\ _p(G)} - b < 1 \tag{2.12}$$

Although (2.12) is only an approximate (possibly highly conservative) solution to model error allocation, we propose to investigate conditions under which it is appropriate. This will provide a useful first step in the study. In particular, we will also consider simple types of nonlinear systems, such as P_{NL} of (2.9) which has a single slope restricted memoryless nonlinearity. After studying that simple form we will examine more complicated cases involving interconnections of slope restricted memoryless nonlinearities.

REFERENCES

[Abr.1] Abrams, C.R., A Design Criterion for Highly Augmented Fly-by-Wire Aircraft,' AIAA Paper No. 82-1570.

[AFWAL] AFWAL Integrated Controls Initiative: Government/Industry Kick-Off Meeting, Wright-Patterson Air Force Base, Ohio, April 16, 1981.

[Des.1] Desoer, C.A , and Vidyasagar, M., Feedback Systems: Input-Output Properties, Academic Press, New York, 1975.

[Des.2] Desoer, C.A. and Liu, R.W., Global Parameterization of Feedback Systems with Nonlinear Plants ' Systems and Control Letters, Vol. 1, No. 4, Jan. 1982.

[Doy 1] Doyle, J.C , and Stein, G., 'Multivariable Feedback Design: Concepts for a Modern/Classical Synthesis,' IEEE Trans. Aut. Contr., AC-26, No. 1, pp. 4-17, February 1981.

[Doy.2] Doyle, J.C., Wall, J.E., and Stein, G., 'Performance and Robustness Analysis for Structured Uncertainty,' Proc. IEEE Conf. on Dec. and Contr., Dec. 1982, Orlando, Florida.

[Eyk.1] Eykhoff, P., System Identification Wiley, 1974.

[Gup.1] Gupta, N K., Hall. W.E., and Trankle, T.L., Advanced Methods for Model Structure Determination from Test Data,' J. Guidance and Control, Vol. 1, No. 3, pp. 197-204, May-June 1978.

[Gup 2] Gupta, N.K., 'Frequency-Shaped Cost Functionals: Extensions of Linear-Quadratic-Gaussian Design Methods,' AIAA J. of Guid. and Control, Nov./Dec. 1980, pp. 529-535.

[Hor 1] Horowitz, I.M., Synthesis of Feedback Systems, Academic Press 1963.

[Kos.1] Kosut, R.L., 'Analysis of Performance Robustness for Uncertain Multivariable Systems,' Proc. IEEE Conf. on Dec. and Contr., Orlando, Florida, December 1982, pp. 1289-1294.

[Kos.2] Kosut, R.L., Salzwedel, H., and Emami-Naeini, A., 'Robust Control of Flexible Spacecraft,' AIAA Journal of Guid. and Control, Vol. 6, No. 2, pp. 104-111, March-April 1983.

[Kos.3] Kosut, R.L., and Friedlander, B., 'Performance Robustness Properties of Adaptive Control Systems ' Proc. 21st IEEE CDC, Orlando, Florida, December 1982.

[Kos.4] Kosut, R.L , 'Robust Adaptive Control: Conditions for Local Stability,' Proc. 1983 ACC, San Francisco, CA, June 1983.

[Saf.1] Safonov, M.G., Stability and Robustness of Multivariable Systems, MIT Press, 1980.

[Saf.2] Safonov, M.G., 'Stability Margins of Diagonally Perturbed Multivariable Feedback Systems,' Proc. 20th IEEE CDC, San Diego, CA, Dec. 1981, pp. 1472-1478.

[Saf.3] Safonov, M.G., 'Tight Bounds on the Response of Multivariable Systems with Component Uncertainty,' Proc. 16th Allerton Conf., Oct. 1978.

[Sil.1] Siljak, D.D., Large-Scale Dynamic Systems: Stability and Structure, New York: North-Holland, 1978.

[Tob.1] Tobie, H.N., Elliott, E.M., and Malcom, L.G., 'A New Longitudinal Handling Qualities Criterion,' Nat. Aerospace Electronics Conf., Ohio, May 16-18, 1966.

[Vid.1] Vidyasagar, M., Input-Output Analysis of Large Scale Interconnected Systems, Springer-Verlag, New York, 1981.

[Zam.1] Zames, G., 'On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems,' IEEE Trans. Aut. Contr., Part I: Vol. AC-11, No. 2, pp. 228-238, April 1966; Part II: Vol. AC-11, No. 3, pp. 465-476, July 1966.

END

FILMED

1-86

DTIC